



Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey

JON PEREZ-CERROLAZA, Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain

JAUME ABELLA, Barcelona Supercomputing Center (BSC), Spain

MARKUS BORG, RISE Research Institutes of Sweden AB, Sweden

CARLO DONZELLA, Exida, Italy

JESÚS CERQUIDES, Artificial Intelligence Research Institute (IIIA-CSIC), Spain

FRANCISCO J. CAZORLA, BSC and Maspatechnologies S.L., Spain

CRISTOFER ENGLUND, RISE Research Institutes of Sweden AB, Sweden

MARKUS TAUBER, Research Studios Austria, Austria

GEORGE NIKOLAKOPOULOS, Luleå University of Technology, Sweden

JOSE LUIS FLORES, Ikerlan Technology Research Centre, BRTA, Spain

Artificial Intelligence (AI) can enable the development of next-generation autonomous safety-critical systems in which **Machine Learning (ML)** algorithms learn optimized and safe solutions. AI can also support and assist human safety engineers in developing safety-critical systems. However, reconciling both cutting-edge and state-of-the-art AI technology with safety engineering processes and safety standards is an open challenge that must be addressed before AI can be fully embraced in safety-critical systems. Many works already address this challenge, resulting in a vast and fragmented literature. Focusing on the industrial and transportation domains, this survey structures and analyzes challenges, techniques, and methods for developing AI-based safety-critical systems, from traditional functional safety systems to autonomous systems. AI *trustworthiness* spans several dimensions, such as engineering, ethics and legal, and this survey focuses on the safety engineering dimension.

The findings, conclusions, and opinions expressed herein are solely those of the authors and do not necessarily represent the view(s) of affiliated institutions, organizations or funding bodies.

Authors' addresses: J. Perez-Cerrolaza, Ikerlan Technology Research Centre, Paseo J. M. Arizmediarreta 2, Arasate/Mondragon, Guipuzcoa, 20.500, Spain and Basque Research and Technology Alliance (BRTA), Kurutz Gain Industrialdea 10, Mendaro, Guipuzcoa, 20.850, Spain; e-mail: jmperez@ikerlan.es; J. Abella, Barcelona Supercomputing Center (BSC), Plaça Eusebi Güell, 1-3, Barcelona, Barcelona, 08034, Spain; e-mail: jaume.abella@bsc.es; M. Borg, RISE Research Institutes of Sweden AB for Markus Borg, Schelevägen 17, Lund, Scania, 223 63, Sweden; e-mail: markus.borg@ri.se; C. Donzella, Exida, Rovereto, Italy, carlo.donzella@exida-dev.com; J. Cerquides, Artificial Intelligence Research Institute (IIIA-CSIC), Campus de la UAB, Bellaterra, Bareclona, 08193, Spain; e-mail: cerquide@iiia.csic.es; F. J. Cazorla, Maspatechnologies S.L., Calle Miguel Hernandez 10, p. 11, pta. 1, Barcelona, Barcelona, 08042, Spain; e-mail: francisco.cazorla@bsc.es; C. Englund, RISE Research Institutes of Sweden AB for Cristofer Englund, Gibraltargatan 35, Gothenburg, Västra Götaland, 412 79, Sweden; e-mail: cristofer.englund@ri.se; M. Tauber, Research Studios Austria, Thurngasse 8, Vienna, Vienna, 1090, Austria; e-mail: markus.tauber@researchstudio.at; G. Nikolakopoulos, Luleå University of Technology, Laboratorievägen 14, Luleå, Norrbotten, 971 87, Sweden; e-mail: george.nikolakopoulos@ltu.se; J. Luis Flores, Ikerlan Technology Research Centre, BRTA, Arrasate/Mondragon, Spain; e-mail: jlflores@ikerlan.es.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

0360-0300/2024/04-ART176 \$15.00

<https://doi.org/10.1145/3626314>

CCS Concepts: • Computing methodologies → Artificial intelligence; Machine learning; • Computer systems organization → Dependable and fault-tolerant systems and networks; Robotics; Robotic autonomy; • Hardware → Safety critical systems;

Additional Key Words and Phrases: functional safety, autonomous systems

ACM Reference format:

Jon Perez-Cerrolaza, Jaume Abella, Markus Borg, Carlo Donzella, Jesús Cerquides, Francisco J. Cazorla, Cristofer Englund, Markus Tauber, George Nikolakopoulos, and Jose Luis Flores. 2024. Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey. *ACM Comput. Surv.* 56, 7, Article 176 (April 2024), 40 pages.

<https://doi.org/10.1145/3626314>

1 INTRODUCTION

Artificial Intelligence (AI) is at the core of recent scientific and industrial advances, such as **Autonomous Driving (AD)** [61, 103, 160] and **Unmanned Aerial Vehicles (UAVs)** [182, 268]. AI technology is a cross-domain innovation driver for numerous novel application use cases [140] and embedded intelligence-driven solutions [147, 247]. In some specific high-integrity application scenarios, AI is increasingly “used to support safety-critical decisions where errors can lead to catastrophic and fatal consequences” [51, 137, 208] (e.g., AD [164, 222, 235], railway interlocking [20, 161, 200, 208], aircraft collision avoidance [153], UAVs [71, 240, 241, 268]).

In this line, it is acknowledged that AI is “one of the only technically and economically viable” technologies for developing autonomous systems [147]. Driven by AD and UAV engineering challenges and the associated economic investment, there is a significant research and engineering effort to define novel technical solutions for developing AI-based autonomous systems [103, 107, 164, 219, 222, 235], neatened with the updating and definition of novel safety standards [15, 133, 137, 275] to deal with AI-specific traits. These solutions are also of interest for multiple transportation domains such as avionics [19, 109], railway [19, 20, 200, 208] and automotive [233, 256], and industrial domain applications such as robotics [259] and driverless industrial trucks [134, 141]. In all of these domains, AI technologies can be used to develop both traditional functional safety systems, as well as next-generation autonomous safety-critical systems [33, 137, 147, 280].

However, existing AI software technologies have several generic limitations related to compliance with current safety standards [33, 147]. The most notorious include the “black box” nature of AI solutions causing limitations regarding their explainability and analyzability [3, 51, 104, 235, 268, 282], and compliance limitations concerning software development lifecycle phases, such as specification correctness and completeness, design, testing, verification, and validation [107, 164, 190, 194, 200, 207, 219, 268, 278]. Due to these limitations (challenges), AI techniques have not been recommended for use in safety-critical systems [56, 120, 200]. In fact, nowadays, there are still no structured development approaches, methods and tools with generic acceptance for developing AI-based safety-critical systems [33, 215]. The evolving normative landscape also attests to this with the recent AI [15, 55, 137], **Safety Of The Intended Functionality (SOTIF)** [133] and autonomous systems safety standards [275, 280] that are in development (drafts) or recently published with limited consolidation of industry best practices [33, 89, 147].

These complexities are compounded by a significant fragmentation of the research contributions targeting the use of AI for developing autonomous systems with [267] and without specific safety considerations [190], different safety AI challenges [13, 110, 147], multiple use cases [19, 140], multiple types of AI [87], different lifecycle phases (e.g., design [208, 276], test [53, 114, 116], verification [5, 77, 117]), generic AI solutions (e.g., reinforcement learning [16]) and safety adaptations (e.g., safe reinforcement learning [94]), with references to multiple existing [120, 131] and novel domain-specific safety standards [58, 133, 215, 267, 275, 280, 280].

Product (§4)	Runtime (§5)	Process (§6)
AI System (§4.1) AI Item (§4.2)	Runtime Learning/Adapt. (§5.1)	AI Safety Engineering (§6.2)
Execution Platform (Inference) (§4.3) Tools and Training Platform (§4.4)		Traditional Safety Eng. (§6.1)
Trustworthiness (§7)		
Engineering Dimension (§7.1) Ethical Dimension (§7.2) Legal Dimension (§7.3)		

Fig. 1. Diagram summarizing the structure of this survey.

Trust becomes paramount in paving the way for the industrial development, commercialization and societal adoption of AI-based safety-critical systems such as AD systems [285] and UAVs [268]. AI *trustworthiness* spans several dimensions, such as engineering, ethics and legal, and this survey focuses on the safety engineering dimension. This survey provides an overview and categorization of the vast and fragmented research contributions that target the development of AI-based safety-critical systems for industrial and transportation domains, from traditional **Functional Safety (FuSa)** to autonomous safety-critical systems. This survey targets researchers and safety engineers concerned with the diligent development of AI-based safety-critical systems in a context where the technology novelty leads to a lack of consolidated industry best practices, and available safety standards have little or no consideration for AI technology [80].

Figure 1 provides a graphical representation of the survey structure in which we categorize and summarize selected key research contributions toward using AI technology for (i) the development of AI-based safety-critical systems (*product*) in Section 4, (ii) runtime learning/adaptation of AI-based safety-critical systems (*runtime*) in Section 5, and (iii) the development *process* of safety-critical systems in Section 6. Previous Sections 2 and 3 describe the basic concepts, terminology and taxonomy used in the remainder of this work. Section 7 discusses *trustworthiness* as a multidimensional (e.g., engineering, ethics, legal) and multidisciplinary foundation for developing and adopting AI-based safety-critical systems. Lastly, Section 8 summarizes the overall conclusion and outlines future research directions.

2 BACKGROUND

We next summarize basic concepts and terms used in the survey like AI (Section 2.1), FuSa standards (Section 2.2), and ML properties (Section 2.3). This survey uses existing dependable and secure computing terminology [22], the AI terminology defined in ISO 22989 [138], and the FuSa terminology defined by safety standards IEC 61508-4 [120] and ISO 26262-1 [131]. This survey also integrates terminology from various research fields as described in the referenced survey publications.

2.1 Artificial Intelligence (AI)

As stated in the VDE-AR-E 2842-61 standard, “there is no generally accepted definition of artificial intelligence” [280]. Furthermore, Feldt et al. [87] claim that “there is not even a consensus around what AI is” (referring to the scope of types of algorithms and models). Nonetheless, ISO 22989 provides an “engineering system” oriented definition of AI used in this survey [138]: “set of methods

or automated entities that together build, optimize and apply a model so that the system can, for a given set of predefined tasks, compute predictions, recommendations, or decisions”.

The term *AI safety* [13, 84] is commonly used in the literature to describe techniques and methods that aim to avoid or mitigate the potential harm that developed AI technology applications could produce to humanity. However, within this survey, the term *AI safety* refers to AI-related techniques, processes, and methods that aim to comply with applicable safety standards (Section 2.2, Section 3.3). Thus, this is a narrower and more focused definition.

Finally, **Machine Learning (ML)** is “the art and science of letting computers learn without being explicitly programmed” [112]. It is a subfield of AI that uses algorithms to learn from example training data sets that implicitly specify the intended functionalities, features, rules and constraints. The learning process can be, for instance, *supervised* (using labeled data), *unsupervised* (not using labelled data), *semisupervised* (using both labeled and unlabeled data) and *reinforcement learning* (“a machine learning agent(s) learns through an iterative process by trial and error”) [16, 94, 138]. When the learned ML solution executes on an embedded system (electronics/software implementation with *model parameters*), it performs inferences in which the ML solution provides online actionable outputs based on the inputs provided. Finally, the generic statement that most of the contributions labeled as AI are in fact ML contributions [151] is also extensible to the research contributions analyzed in the scope of the given survey.

2.2 Functional Safety (FuSa) Standards

The development of safety-critical systems follows stringent certification or assessment processes in accordance with generic and domain-specific safety standards defined by national and international standardization organizations (e.g., ISO) and associations (e.g., **Verband Deutscher Elektrotechniker (VDE)**). FuSa is defined as “part of the overall safety” of a system that assures the “freedom from unacceptable risk” [120], through safety functions embedded in programmable electronics systems (electronics/software). IEC 61508 [120] is a reference generic FuSa standard for industrial (e.g., industrial machinery [125], robotics [124], tractors, machinery for agriculture [130]) and ground transportation domains (automotive [131], railway [56]). Notably, FuSa standards from the air transportation domain (e.g., avionics [226, 230], space [206]) “do not consider IEC 61508 as a reference safety standard” [209]. Yet, they also focus on risk mitigation due to failures in safety functions embedded in programmable electronic systems. Further information concerning FuSa standards and associated certification or assessment processes can be found elsewhere [188, 191].

Among all FuSa standards, there is significant variability in terms, definitions, and requirements. For example, IEC 61508 defines the **Safety Integrity Level (SIL)** with a range of discrete values from lowest to highest integrity (SIL1 - SIL4). And the equivalent in the automotive industry is **Automotive Safety Integrity Level (ASIL)** (ASILA - ASILD) and in avionics **Design Assurance Level (DAL)** (DAL E - DAL A). In this survey, we use the generic IEC 61508 as the reference safety standard and take into technical consideration the ground transportation and industrial domains listed above. We also use automotive ISO 26262, given that automotive AD challenges have attracted a significant number of research publications.

For the most critical systems (SIL4, DAL A), “the probability of a dangerous failure is in the range of 10^{-9} per hour of operation, that is, approximately one dangerous failure every 114.155 years” [209]. Thus, the associated error rate is multiple orders of magnitude smaller than the error rate considered excellent for generic AI solutions (e.g., 99% accuracy) [163]. Attaining such an extremely low probability of dangerous failures requires handling *systematic errors* (e.g., human error, tool error) and *random errors* (e.g., memory bit flip) according to strict safety methods, processes, and techniques. FuSa standards are denoted in the survey as traditional because the first

versions were defined decades ago, and the referenced techniques and methods are based on best practices consolidated in the industry over the last decades. Nonetheless, FuSa standards are also updated to accommodate novel and evolving technologies (e.g., ISO 26262-11 for semiconductors technology).

2.3 ML Properties

Due to the intrinsic stochastic nature of ML training and associated epistemic uncertainties [277, 280], the achievable confidence usually depends on “complex hypotheses” [147] related to the different properties of the training and inference input data (e.g., data drift, distribution, correlation), their coverage (e.g., edge/corner cases, hidden variable) and metrics [280]. In this vein, the safety argumentation of systematic errors management is commonly based on high-level AI-related properties adapted to the context of safety systems [147]. For example, as defined by [147]:

- *Auditability*: “Extent to which an independent examination of the development and verification process of the system can be performed”.
- *Data Quality*: “Extent to which data are free of defects and possess desired features”.
- *Explainability/Interpretability*: “Extent to which an ML system can provide an explanation about a decision in a form understandable by a human” (e.g., see surveys [4, 26, 104]).
- *Monitorability*: “Extent to which a system provides information that allows to discriminate a *correct behavior* from an *incorrect behavior*”.
- *Provability*: “Extent to which mathematical guarantees can be provided that some functional or non-functional properties are satisfied” (e.g., formal verification).
- *Robustness*: “Ability of the system to perform its intended function in the presence of: (a) Abnormal inputs (e.g., sensor failure), (b) Unknown inputs (e.g., unspecified conditions)”.

Nonetheless, several research initiatives aim to mitigate this stochastic nature and simplify the safety argument by enforcing deterministic training processes [198]. Furthermore, the ML model implementation can be either deterministic (e.g., a Neural Network (NN) produces the same outputs given the same inputs [280]) or stochastic [66] if the implementation includes techniques that rely on internal random variables.

3 TAXONOMY

This section summarizes the taxonomy used in the survey to classify **Types of AI (TAIs)** (Section 3.1), levels of automation (Section 3.2), heteronomous and autonomous safety standards (Section 3.3), point of application of AI technology (Section 3.4), and AI safety engineering (Section 3.5). This taxonomy aims to provide neutral classification criteria and definitions of terms, reconciling the high variability of terms and concepts from research contributions and safety standards. For instance, the proposed taxonomy can potentially map to domain-specific terms and concepts such as VDE-AR-E2842-61 standard terms [280], e.g., *AI-based system* (“system level”), *AI item* (“AI element”), *AI safety engineering* (‘AI-blueprint’).

3.1 Type of AI (TAI)

There is a lack of consensus about TAIs in the research community [87, 151]. Some works propose as a starting point the “five tribes of AI” [73], on which this section builds on and adds optimization algorithms to classify the TAIs used in referenced research publications within the survey scope.

- (1) *Connectionists* are design learning algorithms based on optimization techniques such as gradient descent, where models are represented as **Neural Networks (NNs)** and specialized **Deep Learning (DL)** models [103, 212] such as **Deep Neural Networks (DNNs)** [181],

Table 1. Types of AI (TAIs) Per Point of Application Analyzed in the Survey

Type of AI (TAI)	Point of Application (PA)		
	Product	Runtime	Process
Analogizers	-	-	[69]
Bayesians	-	-	[5, 86, 92, 93, 118, 148, 149, 158, 159, 252, 281]
Connectionists	[3, 7, 8, 28, 44, 45, 61, 65, 77, 95, 103, 106, 108, 112, 116, 143, 145, 153, 154, 157–160, 160, 171, 177, 180, 181, 204, 205, 212–214, 218, 229, 243, 244, 254, 257, 261, 262, 290, 291]	[150, 173, 201, 262]	[28, 54, 54, 69, 146, 167]
Optimization	[161, 264]	[272]	[79, 79, 98, 208, 273]
Symbolists	[156, 269]	[172, 173]	[24, 69, 99, 143, 144, 168, 172, 178, 203, 236, 283]

Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and autoencoders.

- (2) *Bayesians* are probabilistic outcome-based graphical model representations for probabilistic inference such as Bayesian and Markov networks.
- (3) *Symbolists* are logic-focused algorithms such as rule-based programming (e.g., “always stop in front of a stop sign”), **Constraint Programming (CP)**, decision trees (e.g., random decision forest [269]), fuzzy logic [172] and rational agents [156].
- (4) *Analogizers* are similarity-based classification algorithms (e.g., **Support Vector Machine (SVM)**).
- (5) *Optimization* algorithms aim to discover optimum or satisfactory solutions performing iterative updates and comparison procedures (e.g., **Genetic Algorithm (GA)**).

And as summarized in Table 1 and the white paper on auditable AI systems [33], *connectionist* is the most common TAI embedded in safety-critical systems (*product, runtime*), and it is commonly used in the development *process* (e.g., DL-driven test scenario generation for DL-based *products*).

3.2 Autonomous, Heteronomous, Automation, Automatic, and Collaborative Systems

There is a high diversity of taxonomies to classify autonomous systems and levels of automation, from generic taxonomies [91, 138, 169, 250] to domain-specific taxonomies such as automotive AD [231], avionics [63, 76], railway [119, 121] and robotics [27, 105, 253]. Hence, as for the AI term definition, there is a lack of cross-domain definition consensus for these terms. However, ISO 22989 [138] provides basic generic definitions adaptable to the scope of the survey:

- *Autonomous* systems operate in an “open environment” (e.g., AD systems operate in an “open parameter space in which an infinite number of different traffic situations can occur” [222]) without human-in-the-loop control and supervision (e.g., AD SAE level 5 [231], avionics 3B [76], generic levels 7-10 [250]). As defined by ISO 22989, *autonomy* constitutes the highest level of automation in which “the system is capable of modifying its operating domain or its goals without external intervention, control or oversight” [138].
- The term *heteronomous* system [138] encompasses different levels of automation that must operate in a “(semi-)open environment” with varying degrees of human collaboration, control and supervision, and integrates the generic term “semi-autonomous”. For example,

Table 2. Summary of Selected FuSa, AI Heteronomous, and Autonomous Safety-critical Systems Standards

Domains		Safety Standards			AI standards for safety systems	Reviews / Surveys
	FuSa	Heteronomous	Autonomous			
Transp.	Space	ECSS-Q-ST-30C/40C	-		-	[188, 191]
	Railway	EN 5012x	IEC 62290, IEC 62267		-	[188, 191, 267]
	Avionics	ARP4754, DO-178C	ASTM F3269-21		(ARP6983)	[188, 191][268]
	Automotive	ISO 26262	ISO/PAS 21448	ISO 4804, ISO 5083, (UL 4600)	(ISO/AWI PAS 8800)	[162, 188, 191]
Industrial	Robotics	ISO 10218-1	-		-	[224]
	Mining & earth moving machinery	EN ISO 19014	ISO 17757, ISO 16001, ISO 18758-2		-	-
	Ind. Machinery	ISO 13849-1	(ISO/TR 22100-5), (ISO 3691-4)		-	[14]
	Agriculture	ISO 25119	ISO 10975, ISO 18497		-	-
	Generic	IEC 61508	VDE-AR-E2842-61		(ISO 5469)	[188, 191]

AD SAE levels 1–4 [186, 231], avionics levels 1A-1B-2-3A [76], railway systems **Grade of automation (GoA)** 1–4 [119, 121], and generic levels 2–6 [250]. *Automation/automated* is defined as “pertaining to a process or system that, under specified conditions, functions without human intervention” [138].

- *Automatic* systems operate in a “closed environment” with well-defined safety rules and constraints known at design time [105]. Thus, the system is neither *autonomous* nor *heteronomous*. It simply executes an automation of safety functions without human intervention (e.g., railway interlocking system [161]) in compliance with applicable **FuSa standards**.
- *Collaborative* robot refers to diverse robot-human collaborative working models ranging from *automatic* (e.g., safety-rated monitored stop) to *heteronomous* and *autonomous* working models [126, 224], and combinations of the previous.

3.3 Heteronomous and Autonomous Safety Standards

Table 2 classifies the most relevant FuSa, heteronomous, and autonomous safety standards (draft standards are represented in parentheses and standards that explicitly consider AI technology are underlined), and identifies among the dozens of AI standardization initiatives [55] those that target the development of AI-based safety-critical systems. The recommended “reading map” for AI practitioners/professionals not specialized in safety-critical systems is the reading of generic and automotive domain FuSa (IEC 61508; ISO 26262), heteronomous/autonomous (VDE-AR-E2842-61; ISO/PAS 21448, UL 4600), and AI standards for safety systems (ISO 5469; ISO/AWI PAS 8800).

3.3.1 *Heteronomous Safety Standards*. The development of novel types of safety-related systems, such as **Advanced Driver-Assistance Systems (ADAS)** [190], led to a novel scenario where safety-critical systems could fail even in the absence of an electronic/software failure. For example, the intended safety function fails due to unexpected operating conditions not considered in the perception ML algorithm training [162]. Thus, there was a need for a novel type of safety standards, complementary with FuSa standards, such as the automotive domain SOTIF [133]. For example, the development of an ML algorithm-based safety perception function integrated into a safety ADAS, requires compliance with the associated SOTIF (e.g., ISO/PAS 21448), applicable AI standards (e.g., ISO 5469, ISO/AWI PAS 8800), and the embedded implementation should comply with the associated FuSa standard (e.g., ISO 26262). Some transportation and industrial domains have already defined domain-specific safety standard drafts [224, 240, 267] (e.g., automotive SAE levels 3–4 [135, 136]; mining and earth moving machinery [127, 129], autoguidance systems for tractors and machinery for agriculture [123], highly automated agricultural machines [128], collaborative robots [126], aircraft systems with complex functions [18]). And some of these standards do not

mention or consider AI, as they could potentially be implemented with different technologies. For example, the machinery domain ISO 22100 technical report [141] describes risk reduction approaches for driverless industrial trucks implemented with or without AI technology. But, within the scope of the survey, we only consider the scenarios where the system is developed with AI technology.

3.3.2 Autonomous Safety Standards. The development of autonomous safety systems leads to a novel scenario in which the safety system makes autonomous decisions without human control/supervision in an open environment. For these novel types of safety systems, which can not be developed and certified with previously described standards (only), the automotive industry has defined several specific standards, such as UL 4600 [275]. Regarding industrial domains, some authors provide an overview and review of industrial safety standards [267], such as autonomous machine systems [132] and driverless industrial trucks [134].

Finally, the VDE-AR-E2842-61 [280] (“development of trustworthiness of autonomous/cognitive systems”) is a generic standard (draft) for developing **Autonomous/Cognitive (AC)** systems. This standard combines *SOTIF*, *heteronomous*, and *autonomous* system considerations with AI technology.

3.4 Point of Application, Usage Level (UL), and Class

This Section briefly reconciles research [87] and ISO 5469 standard taxonomies [137] concerning the AI technology usage type, class, and characteristics. The *point of application* taxonomy proposed by Feldt et al. [87] defines both “when” and “on what” an AI technology is applied using three categories that can be adapted to the survey scope as follows.

- (1) *Product*: A safety-critical system (the *product*) relies on offline embedded AI technology to perform one or more safety functions. As summarized in Figure 3, the AI-based safety-critical system is composed of one or more *AI-based systems* that integrate one or more *AI items*. The *AI item* embeds the AI technology in an electronic/software component [76] with required model parameters, and it is deployed and executed on a given *execution platform* (e.g., GPU).
- (2) *Runtime*: The AI-based safety-critical system integrates AI technology with runtime field learning capability (online). A *runtime* can also be considered a *product* variant that integrates *dynamic reconfiguration* (IEC 61508-7 C.3.10) and becomes a “one of a kind” system.
- (3) *Process*: AI technology can support and facilitate the offline development of a safety function (*safety engineering*) in compliance with the techniques, methods and processes required by applicable safety standards. This is applied during the system development process, but the used AI technology itself is not embedded into the system (unlike a *product/runtime*).

The ISO 5469 **Usage Level (UL)** taxonomy [137, 242] classifies the use of AI technology using four basic levels (*A–D*) that can be related to the previously described *point of application* taxonomy. In a *product/runtime*, a safety function can be implemented using AI technology (*A*), or a non-safety-related function that could interfere with safety function(s) (*C*) or be interference-free (*D*). Furthermore, AI technology can also be used in the safety-critical development *process* (*B*). UL *A* and *B* are further classified based on whether the AI performs automated decisions (*A1, B1*) or not (*A2, B2*). Based on this, AI-based diagnostic functions can be classified as *A2* or *C*. And, as a rule of thumb, the UL of AI-items performing autonomous safety functions is *A1*, while AI-items for automatic, heteronomous, and collaborative safety-critical systems may be *A1* or *A2*.

Finally, the ISO 5469 **class I-II-III** taxonomy [137, 242] defines whether a given AI technology can be used for the development of a given safety-critical system (*product/runtime/process*) in compliance with previously described safety standards (see Section 2.2, Section 3.3). *Class I* solutions can be developed and reviewed in compliance with safety standards (e.g., use of formal

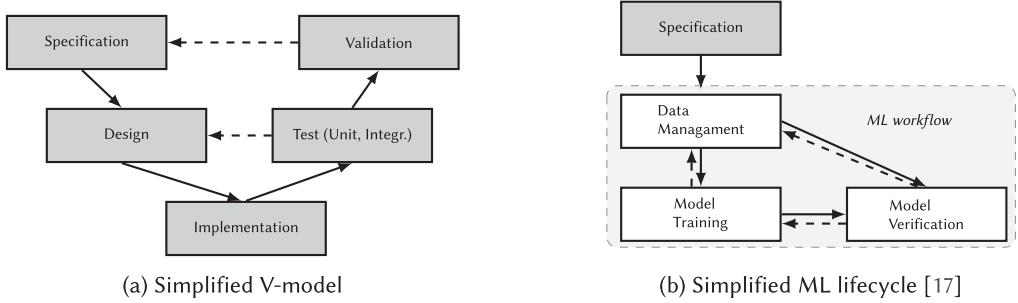


Fig. 2. Simplified lifecycle for *traditional safety engineering* (V-model) and *AI safety engineering* for ML

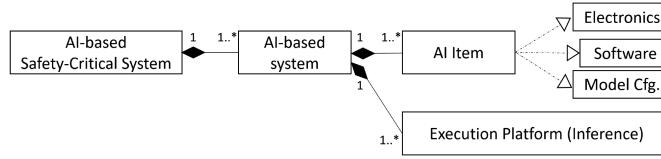
verification [208]). *Class II* solutions cannot be developed and reviewed in compliance with safety standards, but the proposed compensation measures are sufficient for that purpose. For example, the *safety bag/diverse monitor* (C.3.4 IEC 61508-7) technique (a.k.a., run-time checker), safely monitors that the results provided by an AI item are safe [120, 264]. So, the *safety bag* becomes the safety function that prevents unsafe states, and the AI item does not require safety standard compliance. Finally, *Class III* solutions cannot be developed and reviewed in compliance with safety standards, and compensation measures are insufficient. For example, AI-based ADAS using class III AI technology are not considered safety-critical systems, and the driver itself is responsible for driving the vehicle, monitoring the ADAS operation and taking vehicle control in a short time if the ADAS detects and notifies that can no longer provide the intended functionality [66, 163, 285]. And if sufficient compensation measures are defined (e.g., human expert verification, safety bag) a *Class III* solution becomes a *Class II* solution.

3.5 Traditional Safety Engineering and AI Safety Engineering

The *traditional safety engineering* of a safety-critical system follows a V-model development lifecycle as mandated by safety standards (e.g., “realization” phase IEC 61508 [120], “product development” ISO 26262 [131]) with the following generic phases (see Figure 2(a)): specification, design, implementation, **Verification, Validation and Testing** (VVT). The verification activity must confirm that the result of all the development phases (i.e., specification, design, test, and validation) meets the assigned objectives and safety development requirements (IEC 61508-4 Section 3.8.1). And the validation activity must confirm by examination of the evidence (e.g., test results) that the specification has been met (IEC 61508-4 Section 3.8.2) [120].

VDE-AR-E2842-61-1 [280] states that *AI technology* should be considered a third type of technology (in addition to electronics and software) due to its unique characteristics (e.g., uncertainty-related failures). Thus, *AI safety engineering* refers to the engineering lifecycle, processes, activities and techniques required to develop AI-based (sub)systems and AI *items* [215]. The ISO 5469 [137] standard defines a high-level lifecycle that combines the V-model and ML lifecycle activities. Furthermore, the VDE-AR-E2842-61-5 [280] standard states that different TAs might require different processes and lifecycles (still to be defined). For example, while some optimization-based solutions can be developed using a V-model approach [161, 208], most of the analyzed research contributions use a ML workflow [17, 217] or hybrids [174]. Also, Rabe et al. provide an automotive domain specific survey of ML development methodologies [217]. In any case, a relevant difference between *traditional safety engineering* and ML workflows is that the former is specification-driven and the latter data-driven [217].

Figure 2(b) shows the simplified ML lifecycle based on Ashmore et al. [17] used in the survey that, starting from a system *specification* phase [31], follows a ML workflow with *data*

Fig. 3. *Product* composition diagram (UML)Table 3. Selected *product* Safety Techniques (*Class I, II*) and Example Case-studies

Type	UL	Domain	Description	Class	TAI	Technique
<i>Automatic</i>	A	Automotive	Brake pedal state estimation	-	Connectionist	Not specified [8]
		Avionics	Collision avoidance	II	Connectionist	Simulation [153, 154]
		Industrial	Diverse applications	II	Connectionist	Not specified [180]
		Railway	Interlocking system (SIL4)	II	Optimization	Safety bag [161, 264]
	A2, C	Industrial	Sensor diagnostics	-	Connectionist	Diagnostics [145]
<i>Heteronomous and Autonomous</i>	A	Automotive	Collision avoidance (ASIL-D)	II	Connectionist	Safety monitor [7]
		Automotive	Autonomous vehicles platoon	I	Symbolists	Formal verification [156, 269]
			Vehicle collision detection			
		Avionics	AD vehicle overtaking	I	Not specified	Formal verification [223]
			Generic safety pattern for complex functions (e.g., navigation and control)	II	Not specified	Safety monitor [18, 240]
			UAVs and Unmanned Aircraft Systems (UASs)	II	Not specified	Safety monitor [71, 241]
		Industrial	Perception-based solutions for robots		Connectionist	Safety monitor [64]
			Autonomous robots (survey)	I	Not specified	Run-time monitor [218]
		Space	On-board autonomous spacecraft	II	Generic	Formal verification [183]
	A2, C	Automotive	Vehicle self diagnostics	-	Connectionist	Safety bag [40]

management, model learning, and model verification phases. The resulting verified model is then deployed to an execution platform. And the model execution can feed the data management phase with operational data for future model releases.

4 PRODUCT - AI-BASED SAFETY-CRITICAL SYSTEM

This section describes the challenges, techniques, and methods used to develop AI-based safety-critical systems (the *product*) from traditional FuSa to autonomous systems. The description structure follows the *product* layers presented in Section 3.4 and summarized in Figure 3: *AI system* (Section 4.1), *AI item* (Section 4.2), and inference *execution platform* (Section 4.3). We also provide a brief summary of tools and training platforms (Section 4.4).

Table 3 summarizes selected AI safety techniques for the development of AI-based safety-critical *products*. AI technology (*Class I-II*) has already been used for the development of specific FuSa compliant *automatic* safety-critical systems (e.g., SIL4 railway interlocking [161]). Basically, there are two basic approaches for the development of AI-based FuSa systems: the safety verification of all possible input and output combinations either offline using formal verification (*class I*) [278] or online using a safety bag (*class II*) [113, 120, 161, 264]. Regarding AI-based heteronomous and autonomous systems, the generic application of offline formal verification seems questionable due to limitations such as the uncertainty and difficulty of explicitly formalizing all safety specifications, rules and constraints required for the safety verification, and the potential high

Table 4. Product - Summary of Techniques for Systematic and Random Errors Management

Error control and mitigation techniques		AI-based System (Section 4.1)	AI-item (Section 4.2)			
			Connectionist NN	Connectionist DL	Symbolists	Optimization
Systematic Errors	Design Model Training	Safety assurance case [1, 11, 17, 33, 38, 39, 41, 50, 131, 147, 162, 193, 211, 211, 228, 245, 265]	Design and lifecycle [17, 106]		Safety bag, adhoc development [147]	Safety bag [161, 264], adhoc development
			Generic [142, 214, 244, 261, 262, 292]	Safety specific [44, 116, 235, 245, 291]		
	VVT Model Ver.		Safety metrics [117, 294]	Formal methods [117, 294]		
			Metrics [96, 108, 204]			
	Implementation (software, elec.)	VVT [105, 155, 156, 164, 166, 219, 222]	FuSa safety standards compliance (see Section 2.2), e.g., software: IEC 61508-3 7.4.5, 7.4.6			
		Lifecycle [17, 235]	Dataset properties [17, 217]	Engineering requirements [174]	Explicit rules [147, 156]	Safety bag
	Explainability	Uncertainty mgmt. [11, 49, 166, 193, 234, 245, 248, 265, 278]	Generic review [33], Verification [117, 170]	Generic surveys [4, 26, 104]		
			Safety bag, Safety envelope [18, 105, 113, 183, 223, 235, 240, 263]	[18, 64, 71, 240, 241, 268]	Formal ver. [269]	Safety bag [161, 264]
	Monitorability		Formal verification [65, 77, 106, 147, 157, 213, 228, 255, 278]			
	Provability		Test and adversarial attacks [6, 29, 174]			
	Robustness					
Error avoidance, control and mitigation techniques		AI-based System (Section 4.1)	Tools and training platform (Section 4.4)	Execution platform (inference) (Section 4.3)		
ML Properties	Syst. & Random Errors	Safety assurance case [50, 147, 265]	Generic (not qualified) tools and training platforms	Hardware Fwk.	Software Framework	AI Framework
				- Generic dev.: Multicore [196, 202, 210], FPGA [34, 101] GPU [209, 238, 239]	- Generic AD fwk.: e.g., Apollo [10, 256]	- Adapted / Analyzed / Improved: DL [37, 45, 88, 95, 177, 229], basic MxM libraries [88]
Safety standard compliance	FuSa, heteronomous, autonomous and AI & safety standards (see Section 2.2,3,3)			- Specialized dev. [62, 68, 152, 243]: e.g., TPU, NPU, NPU, neuromorphic computing	- Generic: Hypervisor [48, 175, 210]; OS (e.g., Linux [12, 47]); Middlewares [257] (e.g., ROS [183, 187, 256], CyberRT [25, 256], AUTOSAR [21])	- Generic Low level libraries: e.g., TensorRT, OpenBLAS, cuBLAS, ATLAS, cuDNN
				- Custom-designed dev.: e.g., Tesla FSD [258]	- Safety GPU APIs: OpenGL SC, Vulkan SC [209]	- Safety GPU APIs: OpenGL SC, Vulkan SC [209]

dimensional design space that limits the application of formal verification and brute-force testing approaches [33, 183, 268, 278]. A similar limitation applies to online approaches such as the safety bag technique, but in this case, formally specified operational rules can be used to specify safety envelopes (a.k.a., safety monitor, runtime monitor, runtime verification, supervisor, guardian agent, safety layer, safety net) [18, 64, 71, 105, 183, 223, 235, 240, 241, 263]. For example, model checking has already been applied in some specific applications (e.g., AD vehicle overtaking [223]) for the development of formally defined *safety envelope* software (*runtime monitor/verification*) [183, 223].

Safety bag and safety envelope type techniques provide a potentially generic safe approach for the adoption of cutting-edge and state-of-the-art AI technology solutions (as a compensatory measure to adapt *Class III* AI technology to *Class II*). However, its use must consider the safety of the system as a whole because, for example, excessive false alarms could lead to new system-level hazards (e.g., cascade errors in systems with multiple safety functions) and should also consider human cognitive limitations (e.g., cognitive overload, oversight and reaction time limitations) [210, 263]. The avionics domain ASTM F3269 [18] standard describes a reference run-time assurance architecture to safely bound the behavior of “complex functions” integrated in aircraft systems such as UAVs and UASs. This architecture implements a safety bag type technique where a *safety monitor* monitors the safe operation of a “complex function” (e.g., AI-based function) and activates the safe state or switches to a recovery control function [18, 64, 71, 240, 241, 268] if operating outside established safe operation constraints and rules.

In addition, Table 4 summarizes the systematic and random errors management techniques described in this Section. At all levels, the overall AI-based safety-critical must comply with

the required FuSa, heteronomous, autonomous and AI standards. At the highest *AI system* level, developers define safety assurance cases with the arguments and required evidence needed to justify that the system is safe for its purpose; developers identify and manage uncertainty sources and successfully verify, test and validate the system. *AI item* developers control and mitigate systematic errors using at least the appropriate development lifecycle and techniques, appropriate tools and training platforms, and the obtained ML properties provide sufficient evidence to justify the previous assurance case argumentation. Finally, the underlying platform must avoid, control and mitigate systematic and random errors providing sufficient evidence to the previously defined assurance case argumentation.

4.1 AI-based System

Safety assurance cases are commonly used in the development and certification/assessment of traditional FuSa systems to justify that a given safety-critical system is acceptably safe for its purpose, using a structured and evidence-supported safety argumentation [33, 38, 131, 265]. For example, the safety case provides a structured argumentation of systematic and random errors management, from high-level architectural and lifecycle systematic aspects down to the underlying execution platform (see Table 4).

Safety cases are also commonly used for the development and certification/assessment of heteronomous, autonomous, and AI-based safety-critical systems [33, 41, 147, 162, 193, 211, 228, 265]. However, for the latter, the safety assurance case should also support the management of *uncertainty-related failures* (see VDE-AR-E 2842-61 [280]) inherent to heteronomous, autonomous and (non-trivial) AI-based systems. This AI uncertainty management includes, among others, uncertainty sources identification and uncertainty reduction argumentation [248, 265]. For example, the safety assurance case arguments of an *AI-item* (Section 4.2) can be built on claims of high-level properties [17, 147, 228], such as the ML properties defined in Section 2.3 (e.g., *explainability, monitorability, auditability, provability*), arguments based on specific methods used for uncertainty mitigation during the development phases (e.g., data representativeness of requirements, input space coverage validation) [11, 245] and adapt generic argument patterns [211]. However, care must be taken to avoid oversimplifying the safety development challenge to achieving high-level properties with numerical targets and mathematical formulations, without addressing the safety of the system as a whole with associated system hazard elimination [72, 97, 277].

The uncertainty management required to reduce *uncertainty-related failures* becomes a key technical aspect to be managed in all AI-related lifecycle phases from the specification to the verification, validation, and testing phases. For example, in the specification phase of an AI-based heteronomous/autonomous system, the safety functions (and previous safety goals) can only be specified as “intended functionality” with a set of high-level goals and objectives [39], or iterative partial specifications [234], because it is not generally feasible to fully specify the safety functions (w.r.t. all possible scenarios) with a set of safety requirements, rules, constraints (e.g., [32]). This creates a “semantic gap” [49, 193] between the intended functionality and the specified functionality, which sometimes is based on examples where anomalous and edge/corner case examples are a minority. In this context, ensuring that the provided specification provides a correct, accurate and complete representation of the “intended functionality” is a challenge for the *data management and model training* [49]. This challenge can be mitigated by means such as formal verification of safety properties with some degree of uncertainty [278] and safety runtime checkers that during runtime monitor a set of required constraints are always met (safety operational envelope) [166].

On the other hand, the testing and validation of AI-based autonomous systems is still an unsolved key area [67, 111, 166, 268, 284], that limits the practical deployment and commercialization of AI-based safety autonomous systems [67, 155, 164, 166] for which current testing techniques

designed for “manned systems” are not directly applicable and sufficient [266], and field testing only based evidences are generally considered not feasible [155, 222]. Therefore, as described by different authors [155, 156, 164, 166] the validation should consider the definition of a strategy with a framework that combines multiple testing techniques and approaches, with the adaptation of existing techniques and the definition of novel techniques specific for AI-based autonomous systems. In fact, the most relevant challenge in heteronomous and autonomous systems test and validation, is the test and validation of the implemented AI solution itself. So, the most common approach for AI-based AD [155, 166, 219, 222] and collaborative robots [105] testing and validation relies on simulation frameworks where other AI technology solutions facilitate and automatize the process of generating and classifying test scenarios and test cases (see Section 6.2.3).

Finally, the safety case is not static or defined once, as it requires maintenance updates during the system operational life. And this maintenance update requirement is even more crucial for autonomous systems as they operate in complex and continuously evolving environments [33, 50].

4.2 AI Item

This section describes safety technical challenges, techniques, and methods associated with the development of AI-based items using different TAIs abstracted from the application-specific requirements and challenges: *connectionist NN* (Section 4.2.1) and *DL* (Section 4.2.2), *symbolists* (Section 4.2.3), and *optimization* (Section 4.2.4). For all considered TAIs, AI items are implemented as electronics, software, model configuration and combinations of the previous using traditional FuSa standard technical requirements (e.g., IEC 61508-3 software development guidelines) and deployed on execution platforms (see Section 4.3).

4.2.1 Connectionist - Neural Network (NN). At the turn of the millennium, there was growing interest in using NNs in safety-critical applications. In particular, the usage of NNs in aerospace applications and compliance with the stringent aerospace safety standards was an active research area. In this section, we report key aspects to consider when NNs trained using supervised learning enter the picture of safety assurance. Note that the content largely applies also to the subsection on Deep Learning (Section 4.2.2), i.e., NNs for which hidden layers are stacked in attempts to reach human-like performance for perception tasks (e.g., object detection).

Beyond flight controllers, a 2001 review by Lisboa identified a diverse set of industrial use of NNs in safety-related areas [180]. Examples include power generation and transmission, process industries, and transport industries. A common theme among many applications is that NNs were used for automatic control. While **Deep Learning (DL)** has dominated among connectionists in the last decade, (non-deep) NNs remain a valid and useful approach in many applications. Recent examples of NNs within the scope of this article are diagnostics (e.g., sensor error detection [145], vehicle self-diagnostics [290]) and collision avoidance systems in avionics [154].

Companies seeking to integrate NNs in safety-critical systems must evolve several practices throughout the development lifecycle [17, 171, 244]. Supervised learning relies on *data* (for *model training* and *model verification*) being treated as first-class citizens during software and systems engineering. As a result, *data management* needs a rigorous process encompassing collection, augmentation, preprocessing, analysis, and maintenance. *Configuration management* needs to expand to cover the data and feature engineering of the iterative work of NN development. And software *architecture specifications* must also encompass fundamental NN design elements and specifics such as activation functions and hyperparameters controlling the learning process. Furthermore, *specifications* and the associated *test specifications* must be augmented to capture the learning behavior of NNs. Lastly, processes must be adapted to align the highly *iterative development of NNs* with the traditional safety engineering of AI-based systems (V-model).

Concerning *model verification*, Taylor et al. analyzed early research in progress on the VVT of NNs, with a focus on studies relevant for NASA applications [262]. There was substantial research funding assigned to the topic in the early 2000s, and the research matured into several books on the topic, e.g., by Taylor [261], Jacklin et al. [142], Pullum et al. [214], and Schumann and Liu [244]. Menzies and Pecheur provided another early VVT survey in 2005 [194]. While the research was conducted around 20 years ago, the main findings remain relevant today. Discussed challenges of NN VVT include state-space explosion, robustness, explainability, co-engineering of NNs and conventional software, and challenges in specifications of ML concepts. Early VVT solution proposals included “formal methods, control theory, probabilistic methods” [44], and general process frameworks. Again, several ideas from the early era remain relevant, although some do not scale to the DL approaches that will be discussed in Section 4.2.2.

More recently, Zhang and Li provided a systematic literature review [291] of testing and verification techniques for NN software-based safety-critical control systems. This review complements the earlier work through its selection of 83 publications between 2011 and 2018. However, as this time interval coincides with the breakthrough of DL, which Zhang and Li explicitly include, we highlight that the findings partly fit the next subsection of this article – the boundary between NN and DL is not sharp. Based on this analysis, the authors identified five high-order themes, i.e., robustness testing, testing toward failure resilience, measuring test completeness, testing for safety assurance, and testing for explainability. Example solution proposals for NN VVT from the last years include: formal methods [117, 268, 294] and novel dependability metrics [96, 204].

4.2.2 Connectionist - Deep Learning (DL) models. The research community acknowledges the potential benefits of using DL in safety-critical applications. In general, developing safety-critical systems that rely on DL shares the same challenges as NNs – as can be seen in Dey and Lee’s recently proposed three-layered conceptual framework [70]. However, the fact that contemporary deep NNs can be composed of billions of neurons, organized into complex architectures, further amplifies all challenges. Several VTT practices mandated by FuSa become less effective, e.g., code reviews matter less if the logic resides in the training data [235] and the value of adequacy testing metrics is questionable [108].

Still, the representation learning offered by DL has enabled several breakthroughs during the 2010s and trained DL models have outperformed human performance in a range of restricted tasks. From the perspective of this review, the use of DL has disrupted computer vision and enabled perception systems able to generalize to diverse operational contexts. Advances in the automotive industry have been particularly prominent, with DL being a key enabler for AD, and in various ADAS such as automatic emergency braking and lane keeping assistance [28, 61, 160]. Examples of DL use in the aerospace sector include collision avoidance systems [153].

Engineering a trustworthy DL-based system is largely about managing a dynamic *ML workflow* with iterative updates. First, the development of a DL system is an experimental and highly iterative process where the “Changing Anything Changes Everything” principle reigns [246], i.e., all data science activities are intertwined and implications of minor changes are hard to foresee. Second, DL-based systems are typically deployed in dynamic operational environments in which conventional software systems would be insufficient. Third, the AI systems themselves can be dynamic post-release if retraining of internal models is enabled (see Section 5). Thus, integrating automated quality assurance throughout the product lifecycle is essential. Key automation steps, sometimes explained in the context of **ML operations (MLOps)** tools [43, 102], include data version control and experiment tracking to support the iterative DL development and solutions for runtime monitoring [218], e.g., to support detection and management of data drifts.

Model verification explicitly targeting DL-based systems is currently a highly active research topic. Borg et al. provides an automotive domain-specific review of the verification and validation of DL-based solutions [44]. A similar study was reported by Schwalbe and Schels [245]. Zhang et al. found that most academic studies focused on testing the correctness and robustness, while qualities such as interpretability, efficiency, and privacy are much less studied [292]. Riccio et al. concluded in their systematic analysis that test input and test oracle automated generation for DL systems was the most active research topic for DL *model verification* [221]. Huang et al. provided a DNN specific survey [116] covering verification, testing, adversarial attack and defense, and interpretability aspects.

Regarding ML properties for the construction of safety assurance cases, there is a rich variety of research contributions applicable to both NNs and DL models:

- *Data Quality*: The training data implicitly specify the intended functionality, rules and constraints. So data quality is of paramount importance as described by Ashmore et. al [17], and the *data management* phase must produce datasets that exhibit at least properties such as: relevance, completeness, balance, and accuracy [17, 217]. Training data is split for *model training* and *model verification*. In generic applications, the split (e.g., 80%–20%) can be performed randomly, but for safety-critical systems the split shall consider aspects such as: the training data shall completely specify the intended functionality, sufficient representation of edge/corner cases in both training and test data, and the deviation between training/test and operational data shall be minimized [174].
- *Explainability*: Several surveys and reviews summarize the high research activity that addresses the NN and DL models explainability challenge [4, 26, 104, 268]. One can argue that a model is explainable if it is interpretable, and Rudin [227] elaborates on why an interpretable model lowers complexity and thus are to be preferred compared to a model that can not explain the behavior of a NN or DL solution.
- *Provability*: Multiple research contributions address *provability* of NNs and DL models by means of formal verification [65, 77, 106, 147, 157, 213, 228, 255, 278]. However, formal verification is (nowadays) limited to moderate size NNs and certain architectures [147, 157, 268]. For example, the Reluplex method has been used to formally verify ReLu (Rectified Linear Unit) activation properties of a NN with 300 nodes [147, 157].
- *Robustness*: Robustness and resiliency can not be evaluated in the *model verification* with (only) test data [174]. Nonetheless, this is an active research area [29, 174] under the topic of adversarial attacks (security) [6]. The final objective is to analyze and develop solutions that are robust/resilient with respect to (adversarial) perturbations.
- *Auditability*: Huang et al. propose a framework for the automated safety verification of DNNs made classification decisions [117]. Verification is also put forward by Kuper et al. [170] as a viable solution to confirming that NNs behave as intended. In addition, they further suggest to create and use design principles for NNs that produce DNNs that are more amenable to verification [170]. The **European Union (EU)** has proposed an AI act [83] that aims to propose a set of harmonized rules on AI. Hence, the work by Kuper et al. [170], as well as contributions by other scholars presented in this survey, may become building blocks to conform with the proposed AI act.

4.2.3 Symbolists. Decision trees can provide explanations and understandability of decisions made by black-box type AI-based items [104] so that the user is aware of the rationale for decisions and takes control of the safety system if necessary [147] (A2). For example, decision trees can provide runtime explanations of decisions made by an ML-based co-pilot to an aircraft pilot, who

must understand them and react safely in case of wrong decisions [147]. An equivalent approach can be used offline, during the *product* development.

Random forests can also learn safe operation rules from training data to implement safety functions such as vehicle collision detection (*A2*) [269]. Furthermore, whenever feasible, safe operation rules can also be explicitly expressed using formal symbolist languages in *rationale agents* (*A1*) for diverse applications such as autonomous vehicle platooning [156]. Both approaches provide support for *explainability* (white-box), *auditability*, and *provability* (formal verification) requirements. Finally, Törnblom et al. analyze and propose a method and tool for the formal verification of random forests [269].

4.2.4 Optimization. FuSa-compliant optimization algorithm-based safety-critical systems can be developed with the safety bag compensation measure [120] (*Class II*). The optimization function executes a safety related function that is not subject to a complete safety certification process and development, because a run-time safety bag is developed and certified, which ensures that provided results are safe for its purpose and performs associated safety actions if not (e.g., safe state activation). This approach can be used whenever the optimization function cannot be formally verified at design time, or whenever the safety development of optimization software and tools in compliance with FuSa standards requirements is considered not feasible. For example, this safety technique was already used in the 80s to develop a SIL4 railway signaling system that provides optimized and safe results [161, 264].

4.3 Execution Platform (Inference)

The implementation of AI items as embedded software/electronic components with associated model configurations must follow traditional FuSa standard requirements (e.g., software: IEC 61508-3 7.4.5, 7.4.6). Nonetheless, a common approach is to make use of existing execution platforms rather than developing complete ad-hoc implementations. Execution platforms are commonly composed of a hardware platform with **High Performance Computing (HPC)** capability (e.g., **Graphics Processing Unit (GPU)**), a software framework (e.g., hypervisor, AUTOSAR, **Robot Operating System (ROS)**) and an AI software framework (e.g., YOLO, Tensor Flow). And this *execution platform* is the safety computing channel, or one of the safety computing channels of the safety-critical system architecture (e.g., [289]), developed in compliance with applicable FuSa standard requirements. Additionally, in some specific applications, such as AD [258] and UAV systems (e.g., drone) [71, 182], execution platforms must meet **Size, Weight, and Power (SWaP)** constraints while providing the required computing performance and FuSa compliance support [209, 210].

As summarized in the survey by Perez-Cerrolaza et al. [209], the mitigation of random errors by means of evaluation and deployment of diagnostics and fault tolerance mechanisms, is an active research field for DL software frameworks and high-performance computing devices such as GPUs [238, 239], FPGAs [34, 101], multi-core devices [196, 202, 210] and specialized accelerators (e.g., DNN [177]). Or even the definition of specialized software architectures for the development of DL technology-based safety-critical systems [37] and built-in integration of diagnostics measures in software frameworks [88]. The analysis and error mitigation in the DL algorithms and software implementation is also an active research field [45, 95, 177, 229]. Unlike non-DL software, for which fully deterministic and accurate results are expected, DL items often deliver approximate and stochastic results. Hence, error detection is a key challenge for DL items due to multiple challenges: (i) determining whether a result is fault-free is convoluted for a stochastic item that may use also some random numbers as input and whose intrinsic error rate is non-negligible (e.g., object misclassification rates); and (ii) if the DL item inherits a high-integrity level that cannot be

diminished with item decomposition (e.g., using a non-DL item that inherits safety requirements and relieves the DL item), then diverse redundancy may lead to different fault-free results owing to the source of diversity (e.g., different random numbers, different training data, different order of computation causing different rounding of results).

4.3.1 Hardware Platform. As the computing power required to execute AI algorithms such as DL models continues to increase, their deployment is commonly based on generic HPC devices (e.g., GPUs, FPGAs, multi-core devices) [186], specialized accelerators (e.g., **Tensor Processing Units (TPUs)**, **Network Processing Units (NPUs)**, neuromorphic computing) [62, 68, 152, 243] and custom-designed devices (e.g., Tesla FSD [258]) often including specialized accelerators (e.g., DNN accelerator). With respect to FuSa-compliance, the deployment of safety AI items in generic HPC devices is a feasible approach that needs to take into consideration several technical challenges (e.g., random errors, systematic errors, common cause failures) required by associated FuSa standards (e.g., ISO 26262-11, IEC 61508-3 Annex F), as summarized in the specialized surveys for multi-core devices [210], GPUs [209], and FPGAs [34].

4.3.2 Software Framework. Available research and open-source AD specific software frameworks (e.g., Apollo [10]), have some limitations with respect to FuSa compliance that limit their applicability, owing to their use of middlewares and operating systems easing decoupling by means of interfaces to subscribe services to events at the expense of an abuse of pointers, unobvious control flow, and deep if-conditional nesting [256].

These specialized autonomous AD software frameworks, along with traditional FuSa and autonomous safety-critical systems, can be built using generic software frameworks such as domain-specific middlewares, hypervisors, and **Operating Systems (OSs)** [48, 191, 209, 210]. For example:

- Middlewares and domain-specific standard frameworks, including ROS [183, 187], Apollo’s CyberRT [25], and AUTOSAR [21], enable the development of AD frameworks and the use of HPC platforms. On the one hand, some frameworks such as ROS and CyberRT, used along with different versions of Apollo, ease the implementation of AD frameworks, but are not yet integrated with appropriate hypervisors, use interfaces challenging certification (e.g., abundant use of pointers, including function pointers) [256], and do not provide native time predictability [10]. On the other hand, platforms such as AUTOSAR Adaptive are intended to enable the deployment of automotive systems on HPC platforms, but, to our knowledge, they have not been used yet as part of AD frameworks.
- Virtualization technology (e.g., hypervisors) supported by modern multi-core and GPU devices enable the safety compliant integration of software partitions with even different safety criticality levels [48, 210]. However, to our knowledge, AD frameworks do not yet build on hypervisors, partly because those frameworks require HPC devices that may miss the support needed by hypervisors to effectively implement partitioning. Hypervisor technology is, however, planned to be used in some forthcoming hardware platforms and use cases [175].
- There is an increasing interest in Linux for critical systems (e.g., Automotive Grade Linux) and multiple research and industrial project initiatives aim to enable Linux for the development of safety-critical software [12]. For example, Linux is assessed for space systems including HPC SoCs equipped with ML accelerators [47].

4.3.3 AI Framework. A number of AI frameworks, Keras, Pytorch, TensorFlow, MXNet, Theano and Caffe, are highly popular for generic AI applications. Often, DL models are mapped onto those generic frameworks, which are often selected based on characteristics such as user friendliness (often related to the existence of a high-level API), modularity, efficiency, and the like [257].

Table 5. Selected *runtime* Safety Techniques and Example Case-studies

Type	UL	Domain	Description	Class	TAI	Technique
Automatic, Heteronomous or Autonomous	C	Avionics	Intelligent Flight Control System	II	Connectionist	Safety bag [201, 262]
	A	Avionics	Gas turbine aero engine control Generic adaptative control system	II, II	Connectionist, Symbolist Generic, Connectionist	Safe adaptation [172, 173] Safe adaptation [142]
	A	Aerospace	Adaptative guidance	I, II	Connectionist	Limited adaptation [150]
	A, C	Industrial	ILC-based hydraulic machinery	I, II	Optimization	Limited actuation [272]

AI frameworks may already use primitives for mathematical operations used for DL models, such as Generalized **Matrix-Matrix multiplication** (**MxM**), among others. Those primitives are then instantiated for the specific target platform using platform-specific and/or low-level libraries such as TensorRT, OpenBLAS, cuBLAS, ATLAS, and cuDNN, to name a few.

Whether AI frameworks implementation complies with domain-specific standards relates to the specific implementation of the primitives used. Generally, those implementations do not provide any specific safety support, but some APIs and other works provide alternative implementations with safety requirements in mind for CPUs [88] (e.g., embedded diagnostics) and GPUs [209] (e.g., with specific APIs such as OpenGL SC and Vulkan SC).

4.4 Tools and Training Platform

There is a rich and dynamic variety of generic frameworks (e.g., TensorFlow), infrastructure (e.g., GPU servers, cloud infrastructure) and tools for the development of generic AI solutions (e.g., *model training*) [43, 102, 199]. Nonetheless, these generic solutions were not designed with safety standards compliance requirements such as tool qualification. So, this is a potential source of systematic errors (e.g., tool and process errors) and hardware random errors (e.g., training data corruption, GPU random error during *model training*) not generally addressed in research contributions [102, 274].

5 RUNTIME - AI ONLINE LEARNING/ADAPTATION

This section describes selected techniques and methods for the AI online learning/adaptation of AI-based safety-critical systems (*runtime*). By default, *runtime* adaptation leads to a “one of a kind” safety-critical system instantiation that, if unconstrained, is beyond the scope of current and novel safety standards [142, 165]. For example, in this scenario, an AD system might adapt and learn new behaviors [225] that were not considered, verified, and validated in the offline development and safety certification/assessment process [165]. And this adaptation could even be implemented as continuous [11] and lifelong learning [205]. Thus, the “one of a kind” safety-critical system instantiation may differ from the originally certified/assessed system.

However, as summarized in Table 5, it is feasible to consider constrained AI runtime learning/adaptation approaches (Section 5.1), for which correctness and completeness of all possible variants is considered in the safety-critical system development process and safety certification/assessment.

5.1 Runtime Learning/Adaptation

Table 5 summarizes the most relevant techniques and methods selected from research contributions that focus on AI runtime learning/adaptation approaches for developing dependable or safety-critical systems: safety bag (Section 5.1.1), safe adaptation (Section 5.1.2), limited adaptation (Section 5.1.3), limited force (Section 5.1.4) and “library based offline” (Section 5.1.5). Some selected research contributions describe techniques for developing dependable systems and not explicitly safety-critical systems. However, these techniques are adaptable to safety standard requirements;

thus, this section describes and adapts them. Finally, it is assumed that the implementation of described techniques meets basic safety assumptions [11]: e.g., it is an authorized adaptation, the adaptation has a well-defined process (e.g., trigger command, update time) and implements basic error detection/control measures.

5.1.1 Safety Bag. The previously explained *Class II* safety bag technique (a.k.a., safety monitor), can also ensure that the outputs provided by the AI-item subject to runtime learning/adaptation are safe. As previously explained, the safety bag becomes the safety function and the AI-item becomes a non-safety function (C). For example, the avionics **Intelligent Flight Control System (IFCS)** aims to safely optimize aircraft flight performance with two NNs, one trained offline and the second one while the aircraft is in operation (**Online Learning Neural Network (OLNN)**) [260]. And the system runs two safety monitors, one for each NN, where the OLNN safety monitor checks the safeness of the provided outputs. Another example is AI-generated online trajectory monitor of (slow-dynamic) autonomous systems using techniques such as **Nonlinear Model Predictive Control (NMPC)** [201].

5.1.2 Safe Adaptation. The *safe adaptation* technique requires both the AI-item and the runtime learning/adaptation algorithm to be safety-compliant. This is because both must perform safety functions, safe inference, and safe runtime learning/adaptation. For example, Kurd et al. [172, 173] describe a safety-critical “gas turbine aero engine control” based on a hybrid TAI (*connectionist, fuzzy*) that performs runtime adaptation to provide safe control while safely adapting to the engine degradation and environmental change. Additionally, Jacklin et al. [142] describe challenges and example techniques for the development of safe adaptive control solutions using learning algorithms such as NNs (e.g., learning convergence, speed of learning convergence, learning algorithm stability).

5.1.3 Limited Adaptation. The *limited adaptation* technique safely constraints the internal runtime learning/adaptation, either through a safety compliant adaptation (*Class I*) or a safety bag that checks the adaptation outcome (*Class II*, see Section 5.1.1). For example, Johnson et al. [150] describe using NNs to perform adaptive control of an autonomous launch vehicle guidance system. The system uses an adaptive NN-based error cancellation algorithm to cancel the control error due to differences between the actual vehicle dynamics and the design-time vehicle model, with a “bounded weight update law” that safely constrains the runtime learning/adaptation.

5.1.4 Limited Actuation. The *limited actuation* technique ensures that the AI-item subject to runtime learning/adaptation cannot exceed given dangerous output actuation values (e.g., excessive force, energy, voltage). This could be implemented in different ways, such as design-time constraints (e.g., limited input energy leads by design to limited output energy), AI-based safety function that guarantees a limited actuation (*A1, Class I*) or a safety bag that monitors and ensures that output actuation values are within safe limits (C , *Class II*, see Section 5.1.1).

In particular, the **Iterative Learning Control (ILC)** approach is used in dependable industrial control systems such as robots and machinery. ILC [46] aims to optimize the execution of repetitive tasks by learning from previous executions. For example, Trojaola et al. [272] propose an ILC algorithm for hydraulic machinery systems that can be used online to adapt and learn the compensating force required to reduce overshoot and settling time even with unknown knowledge of the valve dynamics. In this scenario, a runtime monitor can be used to monitor and ensure that the learning/adaptation actuation results are safely limited (e.g., compensatory force, dynamic behavior, settling time [272]).

Table 6. Summary of AI-based Development Assistance Solutions for Safety-critical Systems (*Class I and II*)

Lifecycle (Phase)	Usage Purpose	Type of AI (TAI)
Spec.	Hazard identification	Connectionist (NLP), symbolic (ontologies) and analogizer (CBR) [69]
	SIL evaluation	Symbolic (Fuzzy [203, 236]) Bayesian (DBN [252])
Design	Design optimization	Optimization (ACO, EDA, ILS [98, 208])
Test Validation	Test definition automation for MC/DC coverage	Connectionist (NN [54]) Optimization (GA [79]) Symbolic [99]

(a) Traditional safety engineering (V-model)

Lifecycle (Phase)	Usage Purpose	Type of AI (TAI)
Data Mgmt.	See <i>model verification</i> : test definition automation	
Model training	Design optimization (AutoML)	Reinforcement learning [118, 283] Bayesian [118]
Model Verification	Test definition automation	Connectionist (RNN [146], GAN [167], autoencoder [167]) Bayesian [2, 5, 93, 148, 281] Optimization [9, 30, 75, 197, 273] Symbolists [24, 178]
	Test classification automation	Connectionist (CNN [28], RNN [28]) Symbolic (random forest [168])
	Fault injection	Bayesian [149]
	Rule extraction	Symbolist (fuzzy [172], tree [143, 144])
	Quantify uncertainty	Bayesian [86, 92, 158, 159]

(b) AI safety engineering (for ML)

5.1.5 *Library-Based Offline*. The *library-based offline* technique defined for nonlinear control systems [201] can be translated in the safety-critical domain as a library of possible configurations defined and assessed offline, to which the system can transition during runtime (*Class I*). This is the adaptation of a common approach used in the development of traditional safety-critical systems, where all possible configuration and operational modes are defined and assessed offline (e.g., normal and degraded modes of operation).

6 PROCESS - AI-BASED DEVELOPMENT ASSISTANCE

This Section describes AI-based offline techniques and methods that support and facilitate the *traditional safety engineering* of safety-critical systems (Section 6.1) and the *AI safety engineering* of AI items (Section 6.2). For the latter, developers use AI-based solution(s) to develop *AI item(s)* (e.g., DL-based perception item tested using test scenarios defined with Bayesian optimization).

There is a considerable amount of research contributions proposing AI-based techniques to support and assist non-safety-related software developers [87, 192] (e.g., software test automation [114, 179, 220, 237]). However, these generic contributions (*Class III*) cannot be directly used to develop safety-critical systems because they do not comply with the strict method, process and tool qualification requirements imposed by safety standards. Nevertheless, these contributions could complement traditional methods and techniques that already meet the requirements of safety standards. But, the intended use of these contributions would not yet be safety-related and are considered outside the scope of this survey.

On the other hand, as summarized in Table 6, there are multiple research contributions proposing AI-based solutions to support and assist developers of safety-critical systems. It is worth noting that the amount of research contributions focusing on *AI safety engineering* is higher than those focusing on *traditional safety engineering*, due to the novelty of the challenge posed by the former and the diversity and rich variety of “problems” (challenges) to solve (e.g., model verification). Furthermore, this diversity and rich variety of challenges require the use of a diverse and rich variety of AI-based solutions that cover all TAIs summarized in Section 3.1.

Finally, we should also mention that AI solutions are also commonly integrated into hardware ASIC design tools, FPGA synthesis tools and software compilers [115, 176, 286]. And manufacturers for safety-critical systems already address systematic errors through mass-produced electronic integrated circuits requirements (e.g., IEC 61508-2 Section 7.4.6.1) and tool qualification requirements (e.g., IEC 61508-4 Section 3.2.11, ISO 26262-8 Section 11.4.5/6).

6.1 Traditional Safety Engineering

Research contributions that focus explicitly on *traditional safety engineering* of safety-critical systems are fragmented and scarce (see Table 7a). This Section describes some selected example research contributions following the V-model structure (see Figure 2(a)).

6.1.1 Specification, Design, and Implementation. A single systematic error in the requirements, design or implementation phase could directly lead to a fatal consequence. So, safety standard requirements (e.g., tool qualification) are stricter and research contributions that explicitly target safety-related systems are fragmented and scarce (both *class I* and *class II*). For example:

- Specification: **Natural Language Processing (NLP)** solutions can be used for safety assessment and analysis of textual requirements (e.g., hazard identification [69]) with human safety expert verification of the proposed results as a compensation measure to become *Class II*.
- Design and implementation: Optimization algorithms and formal verification techniques can be combined to facilitate the design of FuSa-compliant safety functions [208]. The optimization algorithm proposes an optimized design for a given criterion, and the formal verification verifies compliance with all applicable safety rules and constraints. To do this, the safety requirements that define the safety rules and constraints are expressed both formally for the formal verification and semi-formally for the optimization process. And, as the result is formally verified (*Class I*), state-of-the-art non-safety related AI software tools, engineers and methods can be used for the design optimization proposal activity.

6.1.2 Verification, Validation and Testing (VVT). Software test automation is an active research area for non-safety related systems [114, 179, 220, 237]. Concerning safety-critical systems, the most relevant challenge addressed is the generation of automated test data and test cases to achieve the level of safety software test coverage requested by safety standards [120], such as the **Modified Condition/Decision Coverage (MC/DC)** percentage levels. AI algorithms can facilitate achieving the recommended 100% MC/DC criteria for software unit test activity (IEC 61508 Table B.2), reducing the safety engineering effort required to perform a detailed analysis of all software code paths and test data combinations that could lead to testing all software code statements and execution branches. To that end, symbolic [99], NN [54], and GA [79] solutions have been proposed for test data generation and the achieved MC/DC value can be potentially verified with **Commercial Off-The-Shelf (COTS)** qualified tools [99] (*class I*).

6.2 AI Safety Engineering

Concerning the *AI safety engineering* of AI-based (sub)systems and items, most research contributions describe ML-based solutions for connectionist-based *products*. So this Section follows the ML workflow described in Section 3.5 and Figure 2(b). As summarized in Table 7b, research contributions that target the data management and model learning phases are scarce, and solutions that target the model verification phase are more abundant specially for the VVT activities of heteronomous/autonomous systems.

6.2.1 Data Management. As stated in the generic survey of software engineering for the development of AI-based systems, “data-related issues are the most recurrent type of challenge” with limited mitigation techniques described in the surveyed articles [192]. This generic statement can also be extended to the safety-critical and AI-based *process niche*, which primarily focus on the automated generation of test data and scenarios as described for *model verification* (see Section 6.2.3).

6.2.2 Model Learning. Automated ML (AutoML) refers to the methods, techniques, and processes that aim to automate the development of ML models [17, 33, 118]. For example, selecting

optimal DL hyperparameters for developing ML models for autonomous driving tasks is time-consuming for engineers. And autoML has been (functionally) evaluated as a successful approach for the design automation of perception tasks ML models, with results that outperformed the ones obtained by trial-error approaches by experienced engineers (higher *accuracy*, less latency) [283]. For that purpose, the autoML can use a variety of technical approaches such as random search, reinforcement learning approaches and Bayesian optimization to explore the design space [118, 283]. However, AutoML-based design space exploration requires higher computational resources than human-guided designs and training infrastructure scalability becomes a technical concern [118]. Also, there is still a lack of both safety standard requirements to guide the autoML systematic error reduction and research contributions proposing methods or techniques in this line.

6.2.3 Model Verification. AI technology also plays a crucial role in the scalability, efficiency and automation of AI-based items/systems' testing and validation *processes* (see Section 4.1). The (pseudo) manual definition of test scenarios and test cases is considered not feasible or scalable for heteronomous/autonomous systems [93, 281]. Ma et al. [185] provide an up-to-date review of AI in the VVT of AD systems, dividing the works into scenario-based testing, formal verification, and fault injection testing. This is an active area of research [217, 221], with a rich variety of TAs that can be used for the automation of these VVT tasks, and associated *model verification* activities:

- *Connectionist* solutions: DL technologies can “discover intricate structures well in high-dimensional data and learn the idea of correct representation of data” [8, 254]. Therefore, they are commonly used for the unsupervised modeling and generation of test scenarios/cases, such as vehicle maneuver modeling using autoencoder and **Generative Adversarial Network (GAN)** solutions [167]. One advantage of this approach is that in both cases, the learned model has been trained to generate trajectories that even the discriminator (for GAN) is not able to distinguish between real life or synthetic trajectories [167]. Another common approach is the generation of test scenarios using RNNs (e.g., accident scenarios [146]) or scenario classification using RNNs and CNNs [28]. Furthermore, the number of scenarios explored can be increased dramatically through the use of deep Q-learning [9].
- *Bayesian* solutions [2, 93, 148, 293] are also commonly used for the unsupervised generation of test data, test cases and test scenarios using the learned probability distribution for the given problem to generate variants. For example, generation of intersection scenes [148] and traffic scenarios [281]. And for a given test scenario, Bayesian optimization can be used to learn from observed system outputs and define test cases that could violate predefined safe operation boundaries [93]. Furthermore, Bayesian techniques can also be used for classification (e.g., a nonparametric Bayesian approach has been used to cluster adversarial policies [60]). Finally, Bayesian solutions have also been proposed for fault injection (e.g., a Bayesian fault injection framework uses “causal and counterfactual reasoning about the behavior under a fault” to find faults/errors) [149].
- *Symbolic* solutions: Ontology-based combination “is an essential approach to generate testing scenarios, which combines scenario entities based on ontology theory for the primary goal of coverage” [24, 178]. And random forests are commonly used for unsupervised test scenario clustering and classification [168].
- *Optimization* solutions: Search techniques have been widely applied for testing [232], for example multiobjective search [30], **Monte Carlo Tree Search (MCTS)** [75], adaptive search [197], and requirements-driven test generation automation with simulated annealing [273]. Finally, Fan et al. [86] and Fisac et al. [90] describe Bayesian model learning solutions via Bayesian NNs or statistical Gaussian processes, which support the optimization and safe control design of adaptable safety-critical systems with control stability and safe limits.

As the available offline computing power continues to increase, the use of statistical testing approaches supported by automated test scenarios/cases generation that obtain sufficient statistical representativeness could be a new approach to explore for AI-systems [147], in analogy to probabilistic WCET [52] and probabilistic testing approaches for Linux-based safety systems [12]. AI technology (*process*) can also be used for the verification of AI-items. For example, *symbolist* trees can be used for rule extraction of RNN-based items for both understandability and verification purposes [143], and *Bayesian* methods are proposed for the uncertainty quantification of DL-based safety applications [86, 92, 158, 159].

7 TRUSTWORTHINESS

As stated in the standard VDE-AR-E2842-61 [215, 280], *trustworthiness* “has not generally accepted definition” at least in the context of AI-based safety-critical systems. Nonetheless, if we analyze in detail the standard VDE-AR-E2842-61 [215, 280], technical reviews in the field of safety and AI [49, 72, 116] and generic AI guidelines (e.g., “Ethics guidelines for trustworthy AI” [82]), we can identify at least three dimensions applicable to AI-based safety-critical systems: engineering (Section 7.1), ethics (Section 7.2) and legal dimensions (Section 7.3). Thus there is a multidisciplinary collaboration requirement to address all trustworthiness dimensions (e.g., engineering, philosophy, ethics, social sciences, law), along with a multi-agent collaboration requirement among all relevant actors such as companies, governments, legislators, regulators, standardization organizations, certification bodies, academia and society in general.

Indeed, the increasing importance of trustworthiness in the development of AI-based safety-critical systems is emphasized in the VDE-AR-E2842-61 standard with the **Trustworthiness Performance Level (TPL)** (TPL 0-4) definition that requires trustworthiness attributes traceability through the AI-based system development activities, design patterns supporting the verification of AI properties, and compliance with specific techniques/measures pending definition details in the current draft [280].

7.1 Engineering Dimension

The engineering dimension must cover at least non-functional properties such as robustness, dependability (reliability, availability, maintainability, safety) [22], and cybersecurity [274, 280]. Previous sections (Sections 4, 5, and 6) have already addressed the safety engineering dimension of AI-based safety-critical systems. And implicitly, to some extent, robustness and dependability aspects relevant to the scope of the given survey. Also note that, the engineering trustworthiness relies on previously described safety assurance cases (see Section 4.1) that provide a structured safety engineering argumentation with associated evidences and risk assessment [41].

Concerning cybersecurity, the life cycle of AI is complex by nature, and it involves several phases such as planning, data management, model training, model evaluation and operation. This represents a vast attack surface that can take place in each phase, posing a threat to both security and safety (“no safety without security”). In the planning stage, developers are a candidate to suffer social attacks that can negatively influence the whole process. The data management and model training processes are the pillars for building models, and poison attacks [78, 216] can impact models in different and relevant aspects, such as accuracy in operation [36]. In order to address these threats it is necessary to plan a defense strategy at two levels: data and people. Concerning information, the defense aims to prevent information stealing and adversarial attacks [35] using strategies such as differential privacy [74], data encryption, adversarial training, standardization and verification of data quality, supply chain, and training process [189, 195, 288], among others [59]. On the other hand, regarding people, awareness and training programs for detecting social manipulations are recommended. Finally, in evaluation and operation, several attacks can take place in different

aspects, such as hardware level attacks [271], adversarial attacks [249], inference attacks [251] and stealing of models [270]. In order to address these threats, system developers can use different prevention techniques, such as feature squeezing, compression, randomness, and multiple parallel AI systems [249, 251, 270, 271, 287].

7.2 Ethical Dimension

Several institutions and committees are currently developing AI ethical guidelines [57, 82] and standards [139], in addition to generic ethical standards for system designs such as IEEE 7000 [122, 285]. Regarding AI-based safety-critical systems, at least two distinct ethical issues must be addressed: *engineering ethics* and *machine ethics* [274].

Engineering ethics is linked to the organization's *safety culture* and associated responsibility and accountability towards the development of such systems [49, 72, 274]. *Engineering ethics* is also linked to the industry, societal, policymaker and regulatory consensus required to adapt the **As Low As Reasonably Practicable (ALARP)** principle to these new types of AI-based safety-critical systems that can potentially provide significant societal benefits (e.g., potential car accidents and fatalities reduction with AD systems [155, 222]) with new risks, e.g., which is the acceptable residual risk? [49]. Moreover, as analyzed by Widen et al. [285] and Koopman et al. [163] for the automotive AD domain, the *safety culture* associated to the *engineering ethics* should also encompass the overall business ethics considering aspects such as cooperation with governments for the definition of safe technology regulations, high safety requirements for road testing and deployment, safe management of tradeoff dilemmas between financial risks and safety risks, marketing-engineering-regulation coherency for delivered autonomy levels (e.g., L2+ [66, 163]) and transparency.

On the other hand, *machine ethics* is associated with the moral and ethical decisions that an AI-based *product/runtime* must make during operation. A rich body of research contributions addresses this challenge in the form of dilemma analysis and experiments [23, 42, 49, 100]. In these dilemmas, the autonomous systems are faced with a catastrophic situation where one or several people are in deadly danger in all possible scenarios, and the autonomous system must make a decision that leads to one of these catastrophic scenarios. The key final question is which catastrophic scenario is considered ethically and morally acceptable. For example, in the "moral machine experiment" [23], millions of people from different countries provided 40 million decision answers to an autonomous vehicle driving morale dilemma in which people of different ages, genders and professions are in deadly danger. The result of these experiments confirmed that cultural variation and other variation sources (e.g., economic) lead to different moral and ethical decision preferences, concluding that there is no single universal preference for *machine ethics*. However, the German ethical guidelines strictly prohibits decisions made on human classifications (e.g., gender, age) [163, 184]. In any case, we should request AI-based safety-critical systems to anticipate and mitigate dangerous situations to avoid such moral dilemmas (e.g., defensive driving strategies in AD system) [163, 184].

7.3 Legal Dimension

The **European Commission (EU)** artificial intelligence act aims to propose a "regulation laying down the set of harmonized rules on artificial intelligence" [83]. This act establishes that AI-based safety critical systems shall be cataloged as "high risk" systems subject to specific requirements, such as the conformity assessment process involving notified bodies [83, 274]. That means AI-based safety-critical systems shall be certified/assessed according to applicable domain-specific standards. This is a standardization challenge because for that purpose the industry and standardization committees must first define, update and approve applicable safety standards (see

Section 2.2, Section 3.3). This also implies detailed technical challenges such as meeting the *auditability* property to support the certification/assessment. Moreover, additional regulations will impose additional specific technical challenges, such as providing *explainability* [193] to support “the right to obtain an explanation of the decision” made by AI-algorithms (“meaningful information about the logic involved” [81]) on behalf of an individual, as established by the **General Data Protection Regulation (GDPR)** [81].

In addition to this, the legal dimension has also attracted multiple research contributions to address current legal challenges, such as the liability for damages caused by an AI-based *product/runtime* [49, 85, 279]. Although the operation of AI-based *products/runtime* is not yet regulated by specific legislation, legal norms require that the offender causing damage must indemnify (liability), or a “person who is responsible for the actions of the offender” [279]. But, for example, if a level 5 autonomous driving system crashes due to decisions made autonomously by the embedded AI technology, in a situation that the manufacturer could not reasonably have foreseen and with no possibility for the passengers to avoid it, who is liable for the accident? Furthermore, “could artificial intelligence become a legal person” with associated offender liability? [279]. The current recommendation of the European Commission [85] is that AI not be granted the status of a legal person, as existing parties could instead be held liable in tort for the actions of an AI. However, these and many other related issues remain open multidisciplinary challenges [49, 279].

Finally, there is also a multidisciplinary collaboration requirement between the legal and engineering dimension. For example, in AD there is a need to translate traffic rules written in human natural language into safety engineering rules for the development and runtime verification of AI-systems [223]. This is required for both “holding autonomous vehicles legally accountable” and provide formal safety requirements to reduce the probability of systematic errors [223].

8 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This section describes the overall conclusion (Section 8.1) and future research directions (Section 8.2).

8.1 Conclusion

This survey summarizes and categorizes a vast and fragmented literature addressing the usage of AI technology for developing safety-critical systems for the industrial and transportation domains, from traditional functional safety to next-generation autonomous systems. Specific AI technology instantiations that perform *automated decision-making* (A1) have already been used with compensatory measures (e.g., safety bag) for the development and certification of *automatic* safety-critical systems (e.g., railway interlocking [161]). And the use of AI technology for developing specific *heteronomous* safety functions that require human supervision (A2) is also common in the latest ADAS systems. However, there is still a significant pending research effort and challenge to define generic AI methods, techniques and processes for developing AI-based safety-critical systems that cannot offload safety management onto humans or non-AI systems. Moreover, there is still a considerable standardization, industrial and research effort remaining to formalize applicable AI-related safety standards, settle best industry practices and define novel technical approaches. There may be a perception that the generic development and certification/assessment of AI-based *autonomous* safety-critical systems (A1) will be reached soon. However, we could be at the beginning of the Pareto principle, where 20% of the technological development effort has led to 80% technical results, and AI-based autonomy might seem reasonably achievable soon. However, achieving the following required 20% technical advance might require a considerable additional effort (+80%) due to the difficulty of achieving the required extremely low probability of failure,

the necessary systematic capability and providing the supporting evidence as required by present and future safety standards. All in all, we must pave the way toward the development and certification/assessment of AI-based safety-critical systems due to their potential advantages for society and overall industrial interest. So, we expect that the multidisciplinary combination of AI, trustworthiness and safety-critical systems research fields will be an active and vibrant research area for the years to come.

8.2 Future Research Directions

The applicability of AI-technology for developing safety-critical systems leads to multiple, diverse, and multidisciplinary challenges. In this Section, we just summarize a set of relevant future research directions aligned with the scope of the survey.

All in all, it is necessary to define an *AI safety engineering* approach with a comprehensive set of generic techniques, life cycles, methods, and processes [151, 200, 215] that could pave the way toward the compliance of AI technology for developing traditional FuSa, heteronomous and autonomous safety-critical systems (*product, runtime, process*). This is an engineering and academia research challenge with two basic types of contributions: “how things can be done” and “how things should be done” [209, 210]. The former refers to the safety adaptation of generic cutting-edge and state-of-the-art AI technology (adapting *Class III* to *Class I-II*). In contrast, the latter refers to a bottom-up development of AI technology natively defined for developing safety-critical systems (*Class I*). And both of them should take into consideration the iterative and dynamic life cycle of AI-based systems (e.g., collect operational data to update the ML model) in the context of industrial and transportation domain systems with long product lifetimes (e.g., ≥ 30 years [209]).

As the ML workflow is data-driven, the *data management* must ensure the appropriate *data quality* (e.g., edge/corner cases, data distributional drift) for the safe *model training and verification*. Data must provide a complete, correct and representative specification of the intended safety functionalities, rules and constraints. *Data management* has recurrent challenges and limited research contributions. The systematic error management of *model training* (e.g., AutoML) is also vital for developing safe models, but limited research addresses this challenge. So, both are future research areas with potentially high impact and interest. Not only from a pure AI safety perspective but also from a safety system perspective (e.g., model human driving vs. autonomous driving to better identify representative edge cases and simulation scenarios).

Model verification is an active research area where AI technology is commonly used for the verification *process* of AI-based safety-critical systems (*product, runtime*). There are multiple challenges (e.g., test scenarios/case/generation, test classification) and problems to be solved in order to provide technically compliant and economically efficient solutions for the VVT of AI-based safety-critical systems.

System-level safety assurance cases use ML properties to justify that the system is safe for its purpose (e.g., *explainability, provability, robustness, auditability*). So, research contributions that develop AI technology that natively provides these properties, or contributions that extract, measure and verify these properties become crucial. All properties are important, but *explainability* is critical. From a safety engineering perspective, *explainability* is a pivotal attribute in supporting an AI item’s understandability, verifiability, and auditability. And from a trustworthiness perspective, it is foundational to support the “right to obtain an explanation” and support legal liability analyses providing explainability information for different actors (e.g., engineer, lawyer).

The training tools and platforms on which data is stored, and ML models are trained and verified, are typically based on state-of-the-art solutions with limited or no support for safety systems development (e.g., cloud computing) and non-qualified tools. While academia can provide research contributions, this challenge will likely require an industrial engineering solution.

Additionally, inference execution platforms are an active research area for HPC devices, AI frameworks, and middlewares. The avoidance, control and mitigation of random hardware failures and systematic failures, along with the spatial and temporal independence of execution, are common challenges that such execution platforms must address (e.g., diagnostics, temporal predictability). While generic computing devices [209, 210] are already addressing these challenges, specialized devices (e.g., TPU) and AI frameworks still have limited support. Furthermore, there are multiple specialized future research challenges, such as portability and distribution of models among redundant and diverse computing platforms (e.g., FPGA and GPU) [210].

Finally, trustworthiness leads us to multiple, multidimensional and multidisciplinary future research directions combining engineering, law and ethics disciplines, among others. For example, engineering and machine ethics, liability considerations, explainability for different actors, analysis of human vs. autonomous system behaviors.

ACKNOWLEDGMENTS

We thank the reviewers for taking the time and effort to review the manuscript and provide valuable comments and suggestions, which helped us improve the manuscript.

REFERENCES

- [1] Rusul Abduljabbar, Hussein Dia, Sohani Liyanage, and Saeed Asadi Bagloee. 2019. Applications of artificial intelligence in transport: An overview. *Sustainability* 11, 1 (2019), 189. Retrieved from <https://www.mdpi.com/2071-1050/11/1/189>
- [2] Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. 2019. Generating adversarial driving scenarios in high-fidelity simulators. In *Proceedings of the International Conference on Robotics and Automation*. 8271–8277. DOI: <https://doi.org/10.1109/ICRA.2019.8793740> ISSN: 2577-087X.
- [3] Evan Ackerman. 2017. How Drive.ai is mastering autonomous driving with deep learning > deep learning from the ground up helps drive's cars handle the challenges of autonomous driving. *IEEE Spectrum* (2017). <https://spectrum.ieee.org/how-driveai-is-mastering-autonomous-driving-with-deep-learning>
- [4] A. Adadi and M. Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. DOI: <https://doi.org/10.1109/ACCESS.2018.2870052>
- [5] Prithvi Akella, Ugo Rosolia, Andrew Singletary, and Aaron D Ames. 2020. Formal verification of safety critical autonomous systems via bayesian optimization. arXiv:2009.12909. Retrieved from <https://arxiv.org/abs/2009.12909>
- [6] N. Akhtar and A. Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430. DOI: <https://doi.org/10.1109/ACCESS.2018.2807385>
- [7] Fadi Al-Khoury. 2017. *Safety of Machine Learning Systems in Autonomous Driving*. Thesis.
- [8] Mohammad Al-Sharman, David Murdoch, Dongpu Cao, Chen Lv, Yahya Zweiri, Derek Rayside, and William Melek. 2021. A sensorless state estimation for a safety-oriented cyber-physical system in urban driving: Deep learning approach. *IEEE/CAA Journal of Automatica Sinica* 8, 1 (2021), 169–178. DOI: <https://doi.org/10.1109/JAS.2020.1003474>
- [9] Berat Mert Albaba and Yildiray Yildiz. 2022. Driver modeling through deep reinforcement learning and behavioral game theory. *IEEE Transactions on Control Systems Technology* 30, 2 (2022), 885–892. DOI: <https://doi.org/10.1109/TCST.2021.3075557>
- [10] M. Alcon, H. Tabani, L. Kosmidis, E. Mezzetti, J. Abella, and F. J. Cazorla. 2020. Timing of autonomous driving software: Problem analysis and prospects for future solutions. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium*. 267–280. DOI: <https://doi.org/10.1109/RTAS48715.2020.000-1>
- [11] Rob Alexander, Hamid Asgari, Rob Ashmore, Andrew Banks, Rajiv Bongirwar, Ben Bradshaw, John Bragg, John Clegg, Jane Fenn, Christopher Harper, David Harvey, Nikita Johnson, Catherine Menon, Roger Rivett, Philippa Ryan, Mark Sujan, Nick Tudor, and Stuart Tushingham. 2020. Safety Assurance Objectives for Autonomous Systems.
- [12] I. Allende, N. M. Guire, J. Perez-Cerrolaza, L. G. Monsalve, J. Petersohn, and R. Obermaisser. 2021. Statistical test coverage for Linux-based next-generation autonomous safety-related systems. *IEEE Access* (2021), 1–1. DOI: <https://doi.org/10.1109/ACCESS.2021.3100125>
- [13] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. *Concrete Problems in AI Safety*. Report.
- [14] Sara Anastasi, Marianna Madonna, and Luigi Monica. 2021. Implications of embedded artificial intelligence - machine learning on safety of machinery. *Procedia Computer Science* 180 (2021), 338–343. DOI: <https://doi.org/10.1016/j.procs.2021.01.171>

- [15] 2023. *ARP6983 (WIP) - Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI*. SAE.
- [16] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38. DOI : <https://doi.org/10.1109/MSP.2017.2743240>
- [17] Rob Ashmore, Radu Calinescu, and Colin Paterson. 2021. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys* 54, 5 (2021), 1–39. DOI : <https://doi.org/10.1145/3453444>
- [18] 2021. *ASTM F3269-21: Standard Practice for Methods to Safely Bound Behavior of Aircraft Systems Containing Complex Functions Using Run-Time Assurance*. ASTM.
- [19] J. Athavale, A. Baldovin, R. Graefe, M. Paulitsch, and R. Rosales. 2020. AI and reliability trends in safety-critical autonomous systems on ground and air. In *Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*. 74–77. DOI : <https://doi.org/10.1109/DSN-W50199.2020.00024>
- [20] J. Athavale, A. Baldovin, and M. Paulitsch. 2020. Trends and functional safety certification strategies for advanced railway automation systems. In *Proceedings of the 2020 IEEE International Reliability Physics Symposium*. 1–7. DOI : <https://doi.org/10.1109/IRPS45951.2020.9129519>
- [21] AUTOSAR 2022. AUTOSAR (AUTOMotive Open System ARchitecture). Retrieved September 30, 2022 from <https://www.autosar.org/>
- [22] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr. 2004. Basic concepts and taxonomy of dependable and secure computing. In *Proceedings of the IEEE Transactions on Dependable and Secure Computing*. 11–33.
- [23] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. 2018. The moral machine experiment. *Nature* 563, 7729 (2018), 59–64. DOI : <https://doi.org/10.1038/s41586-018-0637-6>
- [24] Gerrit Bagschik, Till Menzel, and Markus Maurer. 2018. Ontology based scene creation for the development of automated vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. 1813–1820. DOI : <https://doi.org/10.1109/IVS.2018.8500632> ISSN: 1931-0587.
- [25] Baidu. 2021. Apollo CyberRT framework for Autonomous Driving. Retrieved 30 June 2022 from <https://github.com/storypu/CyberRT>
- [26] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. DOI : <https://doi.org/10.1016/j.inffus.2019.12.012>
- [27] Jenay M. Beer, Arthur D. Fisk, and Wendy A. Rogers. 2014. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of Human-robot Interaction* 3, 2 (2014), 74–99. DOI : <https://doi.org/10.5898/JHRI.3.2.Beer>
- [28] Halil Beglerovic, Thomas Schloemicher, Steffen Metzner, and Martin Horn. 2018. Deep learning applied to scenario classification for lane-keep-assist systems. *Applied Sciences* 8, 12 (2018), 2590.
- [29] Vahid Behzadan and William Hsu. 2019. RL-based method for benchmarking the adversarial resilience and robustness of deep reinforcement learning policies. In *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings* 38. Springer, 314–325.
- [30] Raja Ben Abdessalem, Annibale Panichella, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2018. Testing autonomous cars for feature interaction failures using many-objective search. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 143–154. DOI : <https://doi.org/10.1145/3238147.3238192>
- [31] Nelly Bencomo, Jin L.C. Guo, Rachel Harrison, Hans-Martin Heyn, and Tim Menzies. 2022. The secret to better AI and better software (is requirements engineering). *IEEE Software* 39, 1 (2022), 105–110. DOI : <https://doi.org/10.1109/MS.2021.3118099>
- [32] Carl Bergenhem, Rolf Johansson, Andreas Söderberg, Jonas Nilsson, Jörgen Tryggvesson, Martin Törngren, and Stig Ursing. 2015. How to reach complete safety requirement refinement for autonomous vehicles. In *Proceedings of the Critical Automotive Applicat.: Robustness & Safety*.
- [33] Christian Berghoff, Battista Biggio, Elisa Brummel, Vasilios Danos, Thomas Doms, Heiko Ehrich, Thorsten Ganzevoort, Barbara Hammer, Joachim Iden, Sven Jacob, Heidy Khlaaf, Lars Komrowski, Robert Kröwing, Jan Hendrik Metzen, Matthias Neu, Fabian Petsch, Maximilian Poretschkin, Wojciech Samek, Hendrik Schäbe, Arndt von Twickel, Martin Vechev, and Thomas Wiegand. 2020. Towards auditable AI systems. In *Proceedings of the Auditing AI-Systems: From Basics to Applicat. (Workshop at Fraunhofer Forum)*.
- [34] C. Bernardeschi, L. Cassano, and A. Domenici. 2015. SRAM-based FPGA systems for safety-critical applications: A survey on design standards and proposed methodologies. *Journal of Computer Science and Technology* 30, 2 (2015), 373–390. DOI : <https://doi.org/10.1007/s11390-015-1530-5>

- [35] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*. 1467–1474.
- [36] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331.
- [37] Alessandro Biondi, Federico Nesti, Giorgiomaria Cicero, Daniel Casini, and Giorgio Buttazzo. 2019. A safe, secure, and predictable software architecture for deep learning in safety-critical systems. *IEEE Embedded Systems Letters* 12, 3 (2019), 78–82.
- [38] John Birch, Roger Rivett, Ibrahim Habli, Ben Bradshaw, John Botham, Dave Higham, Peter Jesty, Helen Monkhouse, and Robert Palin. 2013. Safety cases and their role in ISO 26262 functional safety assessment. *Computer Safety, Reliability, and Security: 32nd International Conference, SAFECOMP 2013, Toulouse, France, September 24–27, 2013*. DOI : https://doi.org/10.1007/978-3-642-40793-2_15
- [39] John Birch, David Blackburn, John Botham, Ibrahim Habli, David Higham, Helen Monkhouse, Gareth Price, Norina Ratiu, and Roger Rivett. 2020. *A Structured Argument for Assuring Safety of the Intended Functionality*. 408–414. DOI : https://doi.org/10.1007/978-3-030-55583-2_31
- [40] Jp. Blanquart, S. Fleury, M. Hernek, C. Honvaut, F. Ingrand, J. Poncet, D. Powell, N. Strady-Lécubin, and P. Thévenod. 2004. Software safety supervision on-board autonomous spacecraft. In *Proceedings of the 2nd Embedded Real Time Software Congr.*
- [41] R. Bloomfield, H. Khlaaf, P. Ryan Conmy, and G. Fletcher. 2019. Disruptive innovations and disruptive assurance: Assuring machine learning and autonomy. *Computer* 52, 9 (2019), 82–89. DOI: <https://doi.org/10.1109/MC.2019.2914775>
- [42] Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. 2016. The social dilemma of autonomous vehicles. *Science* 352, 6293 (2016), 1573–1576. DOI : <https://doi.org/10.1126/science.aaf2654>
- [43] Markus Borg. 2022. Agility in software 2.0—notebook interfaces and mlops with buttresses and rebars. In *Proceedings of the International Conference on Lean and Agile Software Development*. Springer, 3–16.
- [44] Wnuk, Boris Duran, Christoffer Levandowski, Shenjian Gao, Yanwen Tan, Henrik Kajser, Henrik Lönn, and Jonas Törnqvist. 2018. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. arXiv:1812.05389. Retrieved from <https://arxiv.org/abs/1812.05389>
- [45] A. Bosio, P. Bernardi, A. Ruospo, and E. Sanchez. 2019. A reliability analysis of a deep neural network. In *Proceedings of the 2019 IEEE Latin American Test Symposium*. 1–6. DOI : <https://doi.org/10.1109/LATW.2019.8704548>
- [46] D. A. Bristow, M. Tharayil, and A. G. Alleyne. 2006. A survey of iterative learning control. *IEEE Control Systems Magazine* 26, 3 (2006), 2039–2114.
- [47] Fredrik C. Bruhn, Nandinbaatar Tsog, Fabian Kunkel, Oskar Flordal, and Ian Troxel. 2020. Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space. *CEAS Space Journal* 12, 4 (2020), 551–564. DOI : <https://doi.org/10.1007/s12567-020-00321-9>
- [48] P. Burgio, M. Bertogna, I. S. Olmedo, P. Gai, A. Marongiu, and M. Sojka. 2016. A software stack for next-generation automotive systems on many-core heterogeneous platforms. In *Euromicro Conf. on Digit. System Des.* (2016), 55–59. DOI : <https://doi.org/10.1109/DSD.2016.84>
- [49] Simon Burton, Ibrahim Habli, Tom Lawton, John McDermid, Phillip Morgan, and Zoe Porter. 2020. Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective. *Artificial Intell.* 279 (2020), 103201. DOI : <https://doi.org/10.1016/j.artint.2019.103201>
- [50] Carmen Cárlan, Barbara Gallina, and Liana Soima. 2021. Safety case maintenance: A systematic literature review. In *Proceedings of the Computer Safety, Reliability, and Security*. Ibrahim Habli, Mark Sujan, and Friedemann Bitsch (Eds.), Springer International Publishing, 115–129.
- [51] Davide Castelvecchi. 2016. Can we open the black box of AI? *Nature* 538, 7623 (2016), 20–23.
- [52] Francisco J. Cazorla, Leonidas Kosmidis, Enrico Mezzetti, Carles Hernandez, Jaume Abella, and Tullio Vardanega. 2019. Probabilistic worst-case timing analysis: Taxonomy and comprehensive survey. *ACM ACM Computing Surveys* 52, 1 (2019), 1–35. DOI : <https://doi.org/10.1145/3301283>
- [53] Ján Čegeň. 2020. Machine learning based test data generation for safety-critical software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1678–1681. DOI : <https://doi.org/10.1145/3368089.3418538>
- [54] Ján Čegeň and K. Rástočný. 2020. Test data generation for MC/DC criterion using reinforcement learning. In *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation Workshops*. 354–357. DOI : <https://doi.org/10.1109/ICSTW50294.2020.00063>
- [55] CENELEC. 2020. *CEN-CENELEC Focus Group Report: RoadMap on Artificial Intelligence (AI)*. Report. CENELEC.
- [56] CENELEC. 2020. EN 50128:2011/A1:2020 - Railway Applications: Communication, signalling and processing systems - Software for railway control and protection systems.
- [57] R. Chatila, K. Firth-Butterfield, J. C. Havens, and K. Karachalios. 2017. The IEEE global initiative for ethical considerations in artificial intelligence and autonomous systems [standards]. *IEEE Robotics & Automation Mag.* 24, 1 (2017), 110–110. DOI : <https://doi.org/10.1109/MRA.2017.2670225>

- [58] Peter Chemweno, Liliane Pintelon, and Wilm Decre. 2020. Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. *Safety Science* 129 (2020), 104832. DOI : <https://doi.org/10.1016/j.ssci.2020.104832>
- [59] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. CoRR abs/1811.03728, (2018). Retrieved from <http://arxiv.org/abs/1811.03728>
- [60] Baiming Chen, Xiang Chen, Qiong Wu, and Liang Li. 2022. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2022), 10333–10342. DOI : <https://doi.org/10.1109/TITS.2021.3091477>
- [61] Long Chen, Shaobo Lin, Xiankai Lu, Dongpu Cao, Hangbin Wu, Chi Guo, Chun Liu, and Fei-Yue Wang. 2021. Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. *IEEE Trans. on Intelligent Transportation Syst.* 22, 6, (2021), 3234–3246.
- [62] Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. 2019. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 2 (2019), 292–308. DOI : <https://doi.org/10.1109/JETCAS.2019.2910232>
- [63] B. Clough. 2002. Metrics, schmetrics! how the heck do you determine a UAV's autonomy anyway. In *Proceedings of the Performance Metrics for Intelligent Systems Workshop*.
- [64] Darren Cofer, Isaac Amundson, Ramachandra Sattigeri, Arjun Passi, Christopher Boggs, Eric Smith, Limei Gilham, Taejoon Byun, and Sanjai Rayadurgam. 2020. Run-Time Assurance for Learning-Enabled Systems. In *NASA Formal Methods: 12th International Symposium, NFM 2020, Moffett Field, CA, USA, May 11-15, 2020, Proceedings*, Springer-Verlag, Moffett Field, CA, 361–368. DOI : https://doi.org/10.1007/978-3-030-55754-6_21
- [65] Davide Corsi, Enrico Marchesini, Alessandro Farinelli, and Paolo Fiorini. 2020. Formal verification for safe deep reinforcement learning in trajectory generation. In *Proceedings of the 2020 Fourth IEEE International Conference on Robotic Computing*. IEEE, 352–359.
- [66] M. L. Cummings and B. Bauchowitz. 2022. Safety implications of variability in autonomous driving assist alerting. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2022), 12039–12049. DOI : <https://doi.org/10.1109/TITS.2021.3109555>
- [67] Werner Dahm. 2010. *Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030*. Air University Press, Air Force Research Institute. DOI : <https://doi.org/10.21236/ADA562237>
- [68] William J. Dally, Yatish Turakhia, and Song Han. 2020. Domain-specific hardware accelerators. *Communications of the ACM* 63, 7 (2020), 48–57. DOI : <https://doi.org/10.1145/3361682>
- [69] O. Daramola, T. Stålhane, I. Omoronyia, and G. Sindre. 2013. Using ontologies and machine learning for hazard identification and safety analysis. In *Proceedings of the Managing Requirements Knowledge*. Springer, Berlin, 117–141. DOI : https://doi.org/10.1007/978-3-642-34419-0_6
- [70] Sangeeta Dey and Seok-Won Lee. 2021. Multilayered review of safety approaches for machine learning-based systems in the days of AI. *Journal of Systems and Software* 176 (2021), 110941. DOI : <https://doi.org/10.1016/j.jss.2021.110941>
- [71] E. T. Dill, S. D. Young, and K. J. Hayhurst. [n. d.]. SAFEGUARD: An assured safety net technology for UAS. In *Proceedings of the IEEE/AIAA 35th Digital Avionics Systems Conference*. 1–10. DOI : <https://doi.org/10.1109/DASC.2016.7778009>
- [72] Roel Dobbe. 2022. System safety and artificial intelligence. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 1584. DOI : <https://doi.org/10.1145/3531146.3533215>
- [73] Pedro Domingos. 2018. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, Inc.
- [74] Min Du, Ruoxi Jia, and Dawn Song. 2020. Robust anomaly detection and backdoor attack detection via differential privacy. In *Proceedings of the International Conference on Learning Representations*. 1–19.
- [75] Peter Du and Katherine Driggs-Campbell. 2019. Finding diverse failure scenarios in autonomous systems using adaptive stress testing. *SAE Int. Journal of Connected and Automated Vehicles* 2, 4 (2019), 241–251. DOI : <https://doi.org/10.4271/12-02-04-0018> Number: 12-02-04-0018
- [76] EASA. 2021. *EASA Concept Paper: First usable guidance for Level 1 machine learning applications - A deliverable of the EASA AI Roadmap*. Report. European Union Aviation safety Agency (EASA).
- [77] Ruediger Ehlers. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *Proceedings of the Automated Technology for Verification and Analysis*.
- [78] Matthias Eicher, Patrick Scharpfenecker, Dieter Ludwig, Felix Friedmann, Florian Netter, and Marius Reuther. 2020. *Process Considerations: A Reliable AI Data Labeling Process*. Technical Report. Incenda AI and TÜV SÜD.
- [79] A. El-Serafy, G. El-Sayed, C. Salama, and A. Wahba. 2015. Enhanced genetic algorithm for MC/DC test data generation. In *Proceedings of the International Symposium on Innovations in Intelligent SysTems and Applications*. 1–8. DOI : <https://doi.org/10.1109/INISTA.2015.7276794>

- [80] Meinhard Erben, Wolf Günther, Tobias Sedlmeier, Dieter Lederer, and Klaus-Jürgen Amsler. 2006. Legal aspects of safety designed software development, especially under european law. In *Proceedings of the 3rd Eur. Embedded Real Time Softw.* 6.
- [81] EU. 2016. Regulation (EU) 2016/679 of the European parliament and of the council - on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).
- [82] EU. 2019. Ethics Guidelines for Trustworthy AI. European Commission - High-Level Expert Group on Artificial Intell. (HLEG AI). Retrieved from <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>
- [83] EU. 2021. *Proposal for a Regulation of the European Parliament and the Council - Laying down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts*. European Commission. Retrieved from <https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:52021PC0206>
- [84] Tom Everitt, Gary Lea, and Marcus Hutter. 2018. AGI safety literature review. In *Proceedings of the 27th Internat. Joint Conf. on Artificial Intell.*. DOI : <https://doi.org/10.24963/ijcai.2018/768>
- [85] Expert Group on Liability and New Technologies. 2019. *Liability for Artificial Intelligence and Other Emerging Digital Technologies*. Report. European Comission.
- [86] David D. Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A. Theodorou. 2020. Bayesian learning-based adaptive control for safety critical systems. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*. IEEE, 4093–4099. DOI : <https://doi.org/10.1109/ICRA40945.2020.9196709>
- [87] R. Feldt, F. G. de Oliveira Neto, and R. Torkar. 2018. Ways of applying artificial intelligence in software engineering. In *Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*. 35–41.
- [88] Javier Fernández, Jon Perez, Irune Agirre, Imanol Allende, Jaume Abella, and Francisco J. Cazorla. 2021. Towards functional safety compliance of matrix-matrix multiplication for machine learning-based autonomous systems. *Journal of Systems Architecture* 121 (2021), 102298.
- [89] Patrik Feth, Rasmus Adler, Takeshi Fukuda, Tasuku Ishigooka, Satoshi Otsuka, Daniel Schneider, Denis Uecker, and Kentaro Yoshimura. 2018. Multi-aspect safety engineering for highly automated driving. In *Computer Safety, Reliability, and Security: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings*. Springer, 59–72.
- [90] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. 2019. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Automat. Control* 64, 7 (2019), 2737–2752. DOI : <https://doi.org/10.1109/TAC.2018.2876389>
- [91] Jörgen Frohm. 2008. *Levels of Automation in Production Systems*. Thesis. DOI : <https://doi.org/10.13140/RG.2.1.2797.7447>
- [92] Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. Dissertation. University of Cambridge.
- [93] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings. 2019. Identification of test cases for automated driving systems using bayesian optimization. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*. 1961–1967. DOI : <https://doi.org/10.1109/ITSC.2019.8917103>
- [94] Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.
- [95] Florian Geißler, Syed Qutub, Sayanta Roychowdhury, Ali Asgari, Yang Peng, Akash Dhamasia, Karthik Pattabiraman, and Michael Paulitsch. 2021. Towards a safety case for hardware fault tolerance in convolutional neural networks using activation range supervision. In *IJCAI Workshop on Artificial Intell. Safety (AISafety)*.
- [96] M. Gharib and A. Bondavalli. On the evaluation measures for machine learning algorithms for safety-critical systems. In *15th European Dependable Computing Conference (EDCC)*. 141–144. DOI : <https://doi.org/10.1109/EDCC.2019.00035>
- [97] Mohamad Gharib, Tommaso Zoppi, and Andrea Bondavalli. 2021. Understanding the properness of incorporating machine learning algorithms in safety-critical systems. *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 232–234. DOI : <https://doi.org/10.1145/3412841.3442074>
- [98] Youcef Gheraibia, Khaoula Djafri, and Habiba Krimou. 2018. Ant colony algorithm for automotive safety integrity level allocation. *Applied Intelligence* 48, 3 (2018), 555–569. DOI : <https://doi.org/10.1007/s10489-017-1000-6>
- [99] Sangharatna Godbole, Joxan Jaffar, Rasool Maghareh, and Arpita Dutta. 2021. Toward optimal MC/DC test case generation. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 505–516. DOI : <https://doi.org/10.1145/3460319.3464841>
- [100] Noah J. Goodall. 2014. Ethical decision making during automated vehicle crashes. *Transportation Research Record* 2424, 1 (2014), 58–65.
- [101] E. Grade, A. Hayek, and J. Börcsök. 2016. Implementation of a fault-tolerant system using safety-related Xilinx tools conforming to the standard IEC 61508. In *Proceedings of the 2016 International Conference on System Reliability and Science*. 78–83. DOI : <https://doi.org/10.1109/ICSRS.2016.7815842>

- [102] Tuomas Granlund, Vlad Stirbu, and Tommi Mikkonen. 2021. Towards regulatory-compliant MLOps: Oravizio's journey from a machine learning experiment to a deployed certified medical product. *SN Computer Science* 2, 5 (2021), 1–14.
- [103] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* 37, 3 (2020), 362–386. DOI : <https://doi.org/10.1002/rob.21918>
- [104] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys* 51, 5 (2018), 1–42. DOI : <https://doi.org/10.1145/3236009>
- [105] Jérémie Guiochet, Mathilde Machin, and Hélène Waeselynck. 2017. Safety-critical advanced robots: A survey. *Robotics and Autonomous Systems* 94 (2017), 43–52. DOI : <https://doi.org/10.1016/j.robot.2017.04.004>
- [106] G. Hains, A. Jakobsson, and Y. Khmelevsky. 2018. Towards formal methods and software engineering for deep learning: Security, safety and productivity for DL systems development. In *Proceedings of the 2018 Annual IEEE International Systems Conference*. 1–5. DOI : <https://doi.org/10.1109/SYSCON.2018.8369576>
- [107] David J. Hand and Shakeel Khan. 2020. Validating and verifying AI systems. *Patterns* 1, 3 (2020), 1–3. DOI : <https://doi.org/10.1016/j.patter.2020.100037>
- [108] Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, and Miryung Kim. 2020. Is neuron coverage a meaningful measure for testing deep neural networks?. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 851–862.
- [109] L. H. Harrison, P. J. Saunders, and P. J. Saraceni. 1993. Artificial intelligence and expert systems for avionics. In *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*. 167–172. DOI : <https://doi.org/10.1109/DASC.1993.283552>
- [110] Hongmei He, John Gray, Angelo Cangelosi, Qinggang Meng, T. McGinnity, and Jorn Mehnen. [2020]. The challenges and opportunities of artificial intelligence in implementing trustworthy robotics and autonomous systems. In *Proceedings of the 3rd International Conference on Intelligent Robotic and Control Engineering*. University of Oxford. DOI : <https://doi.org/10.1109/IRCE50905.2020.9199244>
- [111] Philip Helle, Wladimir Schamai, and Carsten Strobel. 2016. Testing of autonomous systems - challenges and current state-of-the-art. In *Proceedings of the INCOSE International Symposium*. Wiley Online Library, 571–584. DOI : <https://doi.org/10.1002/j.2334-5837.2016.00179.x>
- [112] Jens Henriksson, Markus Borg, and Cristofer Englund. 2018. Automotive safety and machine learning: Initial results from a study on how to adapt the ISO 26262 safety standard. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*. 47–49. DOI : <https://doi.org/10.1145/3194085.3194090>
- [113] Jens Henriksson, Christian Berger, Markus Borg, Lars Tornberg, Sankar Raman Sathyamoorthy, and Cristofer Englund. 2021. Performance analysis of out-of-distribution detection on trained neural networks. *Information and Software Technology* 130 (2021), 106409. DOI : <https://doi.org/10.1016/j.infsof.2020.106409>
- [114] H. Hourani, A. Hammad, and M. Lafi. 2019. The impact of artificial intelligence on software testing. In *Proceedings of the IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology*. 565–570. DOI : <https://doi.org/10.1109/JEEIT.2019.8717439>
- [115] Guyue Huang, Jingbo Hu, Yifan He, Jialong Liu, Mingyuan Ma, Zhaoyang Shen, Juejian Wu, Yuanfan Xu, Hengrui Zhang, Kai Zhong, Xuefei Ning, Yuzhe Ma, Haoyu Yang, Bei Yu, Huazhong Yang, and Yu Wang. 2021. Machine learning for electronic design automation: A survey. *ACM Transactions on Design Automation of Electronic Systems* 26, 5 (2021), 1–46. DOI : <https://doi.org/10.1145/3451179>
- [116] Xiaowei Huang, D. Kroening, Wenjie Ruan, Youcheng Sun, Emese Thamo, M. Wu, and Xinping Yi. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review* 37 (2020), 100270.
- [117] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. 2017. Safety verification of deep neural networks. *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24–28, 2017, Proceedings, Part I* 30. Springer Internat. Publishing, 3–29.
- [118] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated Machine Learning - Methods, Systems, Challenges*. SpringerNature. DOI : <https://doi.org/10.1007/978-3-030-05318-5>
- [119] IEC 2009. IEC 62267: Railway applications - Automated urban guided transport (AUGT) - Safety requirements.
- [120] IEC 2010. IEC 61508(-1/7): Functional safety of electrical/electronic/programmable electronic safety-related systems.
- [121] IEC 2014. IEC 62290-1: Railway applications - Urban guided transport management and command/control systems - Part 1: System principles and fundamental concepts.
- [122] IEEE. 2021. IEEE 7000: IEEE Standard Model Process for Addressing Ethical Concerns during System Design.
- [123] ISO. 2009. ISO 10975: Tractors and machinery for agriculture - Auto-guidance systems for operator-controlled tractors and self-propelled machines - Safety requirements.
- [124] ISO. 2011. ISO 10218-1: Robots and robotic devices - Safety requirements for industrial robots Part 1: Robots.

- [125] ISO. 2015. ISO 13849-1: Safety of machinery Safety-related parts of control systems Part 1: General principles for design.
- [126] ISO. 2016. ISO/TS 15066: Robots and robotic devices Collaborative robots.
- [127] ISO. 2017. ISO 16001: Earth-moving machinery - Object detection systems and visibility aids - Performance requirements and tests.
- [128] ISO. 2018. ISO 18497: Agricultural machinery and tractors - Safety of highly automated agricultural machines - Principles for design.
- [129] ISO. 2018. ISO 18758-2: Mining and earth-moving machinery - Rock drill rigs and rock reinforcement rigs - Part 2: Safety requirements.
- [130] ISO. 2018. ISO 25119: Tractors and machinery for agriculture and forestry - Safety-related parts of control systems.
- [131] ISO 2018. ISO 26262(-1/11) Road vehicles - Functional safety.
- [132] ISO 2019. ISO 17757: Earth-moving machinery and mining Autonomous and semi-autonomous machine system safety.
- [133] ISO. 2019. ISO/PAS 21448: Road vehicles Safety of the intended functionality (SOTIF).
- [134] ISO 2020. ISO 3691-4: Industrial trucks Safety requirements and verification Part 4: Driverless industrial trucks and their systems.
- [135] ISO 2020. ISO/TR 4804 Road vehicles Safety and cybersecurity for automated driving systems Design, verification and validation.
- [136] ISO 2021. ISO/AWI TS 5083 Road vehicles Safety for automated driving systems Design, verification and validation.
- [137] ISO 2021. ISO/IEC AWI TR 5469: Artificial intelligence Functional safety and AI systems (draft).
- [138] ISO. 2021. ISO/IEC DIS 22989: Information technology - Artificial Intelligence - Artificial intelligence concepts and terminology (draft).
- [139] ISO 2021. ISO/IEC DTR 24368 - Information technology - Artificial intelligence Overview of ethical and societal concerns (Draft).
- [140] ISO. 2021. ISO/IEC TR 24030: Information technology Artificial intelligence (AI) Use cases.
- [141] ISO 2021. ISO/TR 22100-5: Safety of machinery - relationship with ISO 12100 - Part 5: Implications of artificial intelligence machine learning.
- [142] Stephen Jacklin, Johann Schumann, Pramod Gupta, Michael Richard, Kurt Guenther, and Fola Soares. Development of advanced verification and validation procedures and tools for the certification of learning systems in aerospace applications. In *Infotech@Aerospace*. DOI : <https://doi.org/10.2514/6.2005-6912>
- [143] H. Jacobsson. 2005. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation* 17, 6 (2005), 1223–1263. DOI : <https://doi.org/10.1162/0899766053630350>
- [144] Henrik Jacobsson. 2006. *Rule Extraction from Recurrent Neural Networks*. Thesis.
- [145] Georg Jäger, Sebastian Zug, and António Casimiro. 2018. Generic sensor failure modeling for cooperative systems. *Sensors* 18, 3 (2018). DOI : <https://doi.org/10.3390/s18030925>
- [146] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder. 2018. Accident scenario generation with recurrent neural networks. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems*. 3340–3345. DOI : <https://doi.org/10.1109/ITSC.2018.8569661>
- [147] Eric Jenn, Alexandre Albore, Franck Mamalet, Grégoire Flandin, Christophe Gabreau, Hervé Delseny, Adrien Gaufrisiau, Hugues Bonnin, Lucian Alecu, and Jérémie Pirard. 2020. Identifying challenges to the certification of machine learning for safety critical systems. In *Proceedings of the 10th European Congress on Embedded Real Time Systems*. 29–31.
- [148] S. Jesenski, J. E. Stelle, F. Schiegg, and J. M. Zöllner. 2019. Generation of scenes in intersections for the validation of highly automated driving functions. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. 502–509. DOI : <https://doi.org/10.1109/IVS.2019.8813776>
- [149] Saurabh Jha, Subho Banerjee, Timothy Tsai, Siva K. S. Hari, Michael B. Sullivan, Zbigniew T. Kalbarczyk, Stephen W. Keckler, and Ravishankar K. Iyer. 2019. ML-based fault injection for autonomous vehicles: A case for bayesian fault injection. In *Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 112–124. DOI : <https://doi.org/10.1109/DSN.2019.00025> ISSN: 1530-0889.
- [150] E. N. Johnson, A. J. Calise, and J. E. Corban. 2001. Adaptive guidance and control for autonomous launch vehicles. In *Proceedings of the 2001 IEEE Aerospace Conference Proceedings*. 2669–2682 vol.6. DOI : <https://doi.org/10.1109/AERO.2001.931288>
- [151] Michael I. Jordan. 2019. Artificial intelligence The revolution hasn't happened yet. *Harvard Data Science Review* 1, 1 (2019). DOI : <https://doi.org/10.1162/99608f92.f06c6e61>
- [152] Norman P. Jouppi et al. 2017. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News* 45, 2 (2017), 1–12. DOI : <https://doi.org/10.1145/3140659.3080246>

- [153] Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. 2019. Deep neural network compression for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics* 42, 3 (2019), 598–608. DOI : <https://doi.org/10.2514/1.G003724>
- [154] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer. 2016. Policy compression for aircraft collision avoidance systems. In *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference*. 1–10. DOI : <https://doi.org/10.1109/DASC.2016.7778091>
- [155] Nidhi Kalra and Susan M. Paddock. 2016. *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* RAND Corporation.
- [156] Maryam Kamali, Louise A Dennis, Owen McAree, Michael Fisher, and Sandor M. Veres. 2017. Formal verification of autonomous vehicle platooning. *Science of Computer Programming* 148 (2017), 88–106.
- [157] Guy Katz et al. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the Computer Aided Verification*. Springer International Publishing, 97–117.
- [158] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. 2015. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv:1511.02680. Retrieved from <https://arxiv.org/abs/1511.02680>
- [159] Alex Kendall and Roberto Cipolla. 2016. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*. IEEE, 4762–4769.
- [160] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. on Intelligent Transportation Syst.* 23, 6 (2021), 4909–4926.
- [161] Peter Klein. 1991. The safety-bag expert system in the electronic railway interlocking system elektra. *Operational Expert System Applications in Europe* 3, 4 (1991), 499–506. DOI : [https://doi.org/10.1016/0957-4174\(91\)90175-E](https://doi.org/10.1016/0957-4174(91)90175-E)
- [162] Philip Koopman, Uma Ferrell, Frank Fratik, and Michael Wagner. 2019. A safety standard approach for fully autonomous vehicles. In *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SAS-SUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings* 38. Springer Internat. Publishing, 326–332.
- [163] Philip Koopman, Benjamin Kuipers, William H. Widen, and Marilyn Wolf. 2021. Ethics, safety, and autonomous vehicles. *Computer* 54 (2021), 28–37. DOI : <https://doi.org/10.1109/MC.2021.3108035>
- [164] Philip Koopman and Michael Wagner. 2016. Challenges in autonomous vehicle testing and validation. *SAE Int. J. Trans. Safety* 4, 1 (2016), 15–24. DOI : <https://doi.org/10.4271/2016-01-0128>
- [165] Philip Koopman and Michael Wagner. 2017. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Mag.* 9, 1 (2017), 90–96. DOI : <https://doi.org/10.1109/MITS.2016.2583491>
- [166] Philip Koopman and Michael Wagner. 2018. *Toward a Framework for Highly Automated Vehicle Safety Validation*. In SAE Tech. Paper 2018-01-1071. DOI : <https://doi.org/10.4271/2018-01-1071>
- [167] R. Krajewski et al. 2018. Data-driven maneuver modeling using generative adversarial networks and variational autoencoders for safety validation of highly automated vehicles. In *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems*. 2383–2390. DOI : <https://doi.org/10.1109/ITSC.2018.8569971>
- [168] F. Kruber et al. 2019. Unsupervised and supervised learning with the random forest algorithm for traffic scenario clustering and classification. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. 2463–2470. DOI : <https://doi.org/10.1109/IVS.2019.8813994>
- [169] Stefan Kugele, Ana Petrovska, and Ilias Gerostathopoulos. 2021. Towards a taxonomy of autonomous systems. In *Proceedings of the 15th European Conference on Software Architecture*.
- [170] Lindsey Kuper, Guy Katz, Justin Gottschlich, Kyle Julian, Clark Barrett, and Mykel Kochenderfer. 2018. Toward scalable verification for safety-critical deep networks. *ArXiV* (2018).
- [171] Zeshan Kurd, Tim Kelly, and Jim Austin. 2007. Developing artificial neural networks for safety critical systems. *Neural Computing and Applications* 16, 1 (2007), 11–19. DOI : <https://doi.org/10.1007/s00521-006-0039-9>
- [172] Zeshan Kurd and Tim P. Kelly. 2004. Using fuzzy self-organising maps for safety critical systems. In *Proceedings of the Reliability Engineering & System Safety*. Maritta Heisel, Peter Liggesmeyer, and Stefan Wittmann (Eds.), Springer, Berlin, 17–30.
- [173] Zeshan Kurd and Tim P. Kelly. 2005. Using safety critical artificial neural networks in gas turbine aero-engine control. In *Proceedings of the International Conference on Computer Safety, Reliability, and Security*. Springer-Verlag, 136–150. DOI : https://doi.org/10.1007/11563228_11
- [174] Hiroshi Kuwajima, Hirotoshi Yasuoka, and Toshihiro Nakae. 2020. Engineering problems in machine learning systems. *Machine Learning* 109, 5 (2020), 1103–1126. DOI : <https://doi.org/10.1007/s10994-020-05872-w>
- [175] Kai Lampka and Adam Lackorzynski. 2019. Using hypervisor technology for safe and secure deployment of high-performance multicore platforms in future vehicles. In *Proceedings of the 2019 26th IEEE International Conference on Electronics, Circuits and Systems*. DOI : <https://doi.org/10.1109/ICECS46596.2019.8964912>

- [176] H. Leather and C. Cummins. 2020. Machine learning in compilers: Past, present and future. In *Proceedings of the Forum for Specification and Design Languages*. 1–8. DOI : <https://doi.org/10.1109/FDL50818.2020.9232934>
- [177] Guanpeng Li et al. 2017. Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Assoc. for Comput. Machinery. DOI : <https://doi.org/10.1145/3126908.3126964>
- [178] Yihao Li, Jianbo Tao, and Franz Wotawa. 2020. Ontology-based test generation for automated and autonomous driving functions. *Information and Software Technology* 117, C (2020). DOI : <https://doi.org/10.1016/j.infsof.2019.106200>
- [179] R. Lima, A. M. R. da Cruz, and J. Ribeiro. 2020. Artificial intelligence applied to software testing: A literature review. In *Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies*. 1–6. DOI : <https://doi.org/10.23919/CISTI49556.2020.9141124>
- [180] Paulo Lisboa. 2001. *Industrial Use of Safety-related Artificial Neural Networks*. Report. Health & Safety Executive (HSE). Retrieved from http://www.hse.gov.uk/research/crr_pdf/2001/crr1327.pdf
- [181] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26. DOI : <https://doi.org/10.1016/j.neucom.2016.12.038>
- [182] Yalin Liu et al. 2020. Unmanned aerial vehicle for internet of everything: Opportunities and challenges. *Computer Communications* 155 (2020), 66–83. DOI : <https://doi.org/10.1016/j.comcom.2020.03.017>
- [183] Matt Luckcuck, Marie Farrell, Louise A. Dennis, Clare Dixon, and Michael Fisher. 2019. Formal specification and verification of autonomous robotic systems: A survey. *ACM Computings Surveys* 52, 5 (2019), 1–41. DOI : <https://doi.org/10.1145/3342355>
- [184] Christoph Lütge. 2017. The german ethics code for automated and connected driving. *Philosophy & Technology* 30, 4 (2017), 547–558. DOI : <https://doi.org/10.1007/s13347-017-0284-0>
- [185] Yining Ma, Chen Sun, Junyi Chen, Dongpu Cao, and Lu Xiong. 2022. Verification and validation methods for decision-making and planning of automated vehicles: A review. *IEEE Transactions on Intelligent Vehicles* 7, 3 (2022), 1–20. DOI : <https://doi.org/10.1109/TIV.2022.3196396>
- [186] Y. Ma, Z. Wang, H. Yang, and L. Yang. 2020. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of Automatica Sinica* 7, 2 (2020), 315–329. DOI : <https://doi.org/10.1109/JAS.2020.1003021>
- [187] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics* 7, 66 (2022), eabm6074. DOI : <https://doi.org/10.1126/scirobotics.abm6074> <https://www.science.org/doi/pdf/10.1126/scirobotics.abm6074>
- [188] Joseph Machrouh, Jean-Paul Blanquart, Philippe Baufreton, J. L. Boulanger, Hervé Delseny, Jean Gassino, Gérard Ladier, Emmanuel Ledinot, Michel Leeman, and Jean-Marc Astruc. 2012. Cross domain comparison of System Assurance. In *Embedded Real Time Software and Systems (ERTS)*.
- [189] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the Int. Conf. on Learning Representations*.
- [190] Klaus Mainzer. 2020. *How Safe Is Artificial Intelligence?* Springer, Berlin, 243–266. DOI : https://doi.org/10.1007/978-3-662-59717-0_11
- [191] I. Martinez et al. 2018. *Safety Certification of Mixed-Criticality Systems*. CRC Press.
- [192] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2022. Software engineering for AI-based systems: A survey. *ACM Trans. Softw. Eng. Methodol.* 31, 2 (2022), 1–59. DOI : <https://doi.org/10.1145/3487043>
- [193] J. McDermid and Yan Jia. 2020. Safety of artificial intelligence: A collaborative model. In *Proceedings of the AISafety@IJCAI*.
- [194] Tim Menzies and Charles Pecheur. 2005. Verification and validation and artificial intelligence *Advances in Computers*. 65 (2005), 153–201. DOI : [https://doi.org/10.1016/S0065-2458\(05\)65004-8](https://doi.org/10.1016/S0065-2458(05)65004-8)
- [195] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. In *5th International Conference on Learning Representations (ICLR)*.
- [196] S. Mittal and J. S. Vetter. 2016. A survey of techniques for modeling and improving reliability of computing systems. *IEEE Transactions on Parallel and Distributed Systems* 27, 4 (2016), 1226–1238. DOI : <https://doi.org/10.1109/TPDS.2015.2426179>
- [197] Galen E. Mullins, Paul G. Stankiewicz, R. Chad Hawthorne, and Satyandra K. Gupta. 2018. Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. *Journal of Systems and Software* 137 (2018), 197–215. DOI : <https://doi.org/10.1016/j.jss.2017.10.031>
- [198] Prabhat Nagarajan et al. 2019. Deterministic Implementations for Reproducibility in Deep Reinforcement Learning. arXiv:1809.05676. Retrieved from <https://arxiv.org/abs/1809.05676>
- [199] Giang Nguyen et al. 2019. Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review* 52, 1 (2019), 77–124. DOI : <https://doi.org/10.1007/s10462-018-09679-z>

- [200] Odd Nordland. 2004. Can artificial intelligence be safe?. In *Proceedings of the Probabilistic Safety Assessment and Management*, Cornelia Spitzer, Ulrich Schmocke, and Vinh N. Dang (Eds.). Springer London, 400–405.
- [201] M. Osborne, H. S. Shin, and A. Tsourdos. 2021. A review of safe online learning for nonlinear control systems. In *Proceedings of the International Conference on Unmanned Aircraft Systems*. 794–803. DOI : <https://doi.org/10.1109/ICUAS51884.2021.9476765>
- [202] M. Ottavi, D. Gizopoulos, and S. Pontarelli. 2018. *Dependable Multicore Architectures at Nanoscale*. Springer. DOI : <https://doi.org/10.1007/978-3-319-54422-9>
- [203] Nouara Ouazraoui and Rachid Nait-Said. 2019. An alternative approach to safety integrity level determination: Results from a case study. *International Journal of Quality & Reliability Management* 36, 10 (2019), 1784–1803. DOI : <https://doi.org/10.1108/IJQRM-02-2019-0065>
- [204] Molly O'Brien, William Goble, Greg Hager, and Julia Bukowski. 2020. Dependable neural networks for safety critical tasks. In *Proceedings of the Engineering Dependable and Secure Machine Learning Systems*. 126–140. DOI : https://doi.org/10.1007/978-3-030-62144-5_10
- [205] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71. DOI : <https://doi.org/10.1016/j.neunet.2019.01.012>
- [206] Joseph N. Pelton and Ram S. Jakhu. 2010. *Space Safety Regulations and Standards*. Butterworth-Heinemann, Oxford. 495 pages. DOI : <https://doi.org/10.1016/B978-1-85617-752-8.10045-5>
- [207] A. Pereira and C. Thomas. 2020. Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction* 2, 4 (2020), 579–602.
- [208] J. Perez, J. L. Flores, C. Blum, J. Cerquides, and A. Abuin. 2022. Optimization techniques and formal verification for the software design of boolean algebra based safety-critical systems. *IEEE Transactions on Industrial Informatics* 18, 1 (2022), 620–630. DOI : <https://doi.org/10.1109/TII.2021.3074394>
- [209] Jon Perez-Cerrolaza, Jaume Abella, Leonidas Kosmidis, Alenjandro J. Calderon, Francisco J. Cazorla, and Jose Luis Flores. 2022. GPU devices for safety-critical systems: A survey. *ACM Computing Surveys* 55, 7 (2022), 1–37. DOI : <https://doi.org/10.1145/3549526>
- [210] Jon Perez Cerrolaza, Roman Obermaisser, Jaume Abella, Francisco J. Cazorla, Kim Grüttner, Irune Agirre, Hamidreza Ahmadian, and Imanol Allende. 2020. Multi-core devices for safety-critical systems: A survey. *ACM Computing Surveys* 53, 4 (2020), 1–38. DOI : <https://doi.org/10.1145/3398665>
- [211] Chiara Picardi, Colin Paterson, Richard David Hawkins, Radu Calinescu, and Ibrahim Habli. 2020. Assurance argument patterns and processes for machine learning in safety-related systems. In *Proceedings of the Workshop on Artificial Intelligence Safety*. 23–30.
- [212] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys* 51, 5 (2018), 1–36. DOI : <https://doi.org/10.1145/3234150>
- [213] Luca Pulina and Armando Tacchella. 2012. Challenging SMT solvers to verify neural networks. *AI Communications* 25, 2 (2012), 117–135.
- [214] Laura Pullum et al. 2007. *Guidance for the Verification and Validation of Neural Networks*. John Wiley & Sons, Inc.
- [215] Henrik J. Putzer et al. 2021. Trustworthy AI-based systems with VDE-AR-E 2842-61. In *Proceedings of the Embedded World*.
- [216] Erwin Quiring et al. 2020. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*.
- [217] M. Rabe et al. 2021. Development methodologies for safety critical machine learning applications in the automotive domain: A survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 129–141. DOI : <https://doi.org/10.1109/CVPRW53098.2021.00023>
- [218] Q. M. Rahman, P. Corke, and F. Dayoub. 2021. Run-time monitoring of machine learning for robotic perception: A survey of emerging trends. *IEEE Access* 9 (2021), 20067–20075. DOI : <https://doi.org/10.1109/ACCESS.2021.3055015>
- [219] N. Rajabli et al. 2021. Software verification and validation of safe autonomous cars: A systematic literature review. *IEEE Access* 9 (2021), 4797–4819. DOI : <https://doi.org/10.1109/ACCESS.2020.3048047>
- [220] Arvind Ramanathan et al. 2016. Integrating symbolic and statistical methods for testing intelligent systems: Applications to machine learning and computer vision. In *Proceedings of the 2016 Design, Automation and Test in Europe Conference & Exhibition*.
- [221] Vincenzo Riccio, Gunel Jahangirova, Andrea Stocco, Nargiz Humbatova, Michael Weiss, and Paolo Tonella. 2020. Testing machine learning based systems: A systematic mapping. *Empirical Software Engineering* 25, 6 (2020), 5193–5254.
- [222] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer. 2020. Survey on scenario-based safety assessment of automated vehicles. *IEEE Access* 8 (2020), 87456–87477. DOI : <https://doi.org/10.1109/ACCESS.2020.2993730>

- [223] Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immller, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. 2017. Formalising and monitoring traffic rules for Autonomous Vehicles in Isabelle/HOL. In *Integrated Formal Methods (IFM)*, 50–66. DOI : https://doi.org/10.1007/978-3-319-66845-1_4
- [224] D. Rodríguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja. 2021. Human-robot interaction review: Challenges and solutions for modern industrial environments. *IEEE Access* 9 (2021), 108557–108578. DOI : <https://doi.org/10.1109/ACCESS.2021.3099287>
- [225] Jurgen Ronald. 2013. *Autonomous Driving – A Practical Roadmap* (2010-01-2335). SAE, 5–26.
- [226] RTCA. 2011. DO-178C/EUROCAE ED-12C - Software Considerations in Airborne Systems and Equipment Certification.
- [227] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. DOI : <https://doi.org/10.1038/s42256-019-0048-x>
- [228] Alexander Rudolph, Stefan Voget, and Jürgen Mottok. 2018. A consistent safety case argumentation for artificial intelligence in safety related automotive systems. In *Proceedings of the European Congr. Embedded Real-Time Syst.* .
- [229] A. Ruospo, A. Bosio, A. Ianne, and E. Sanchez. 2020. Evaluating convolutional neural networks reliability depending on their data representation. In *Proceedings of the 23rd Euromicro Conf. on Digital System Design*. 672–679. DOI : <https://doi.org/10.1109/DSD51259.2020.00109>
- [230] SAE. 2010. Aerospace Recommended Practice ARP4754 Guidelines For Development Of Civil Aircraft and Systems.
- [231] SAE. 2014. J3016 - Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.
- [232] Aneesa Saeed, Siti Hafizah Ab Hamid, and Mumtaz Begum Mustafa. 2016. The experimental applications of search-based techniques for model-based testing: Taxonomy and systematic literature review. *Applied Soft Computing* 49 (2016), 1094–1117. DOI : <https://doi.org/10.1016/j.asoc.2016.08.030>
- [233] Rick Salay and Krzysztof Czarnecki. 2018. Using machine learning safely in automotive software: An assessment and adaption of software process requirements in ISO 26262. arXiv:1808.01614. Retrieved from <https://arxiv.org/abs/1808.01614>
- [234] Rick Salay and Krzysztof Czarnecki. 2019. Improving ML safety with partial specifications. In *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings* 38. Springer, 288–300.
- [235] R. Salay, R. Queiroz, and K. Czarnecki. 2018. An analysis of ISO 26262: Machine learning and safety in automotive. In *WCX World Congress Experience (SAE Technical Paper 2018-01-1075)*, SAE. DOI : <https://doi.org/https://doi.org/10.4271/2018-01-1075>
- [236] Mohamed Sallak, Christophe Simon, and Jean-François Aubry. 2006. Evaluating safety integrity level in presence of uncertainty. In *Proceedings of the 4th International Conference on Safety and Reliability*.
- [237] João Alexandre Pedroso Salvado. 2019. *Artificial Intelligence Applied to Software Testing*. Thesis.
- [238] F. Fernandes dos Santos et al. 2017. Evaluation and mitigation of soft-errors in neural network-based object detection in three GPU architectures. In *Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*. DOI : <https://doi.org/10.1109/DSN-W.2017.47>
- [239] Fernando Fernandes dos Santos, Luigi Carro, and Paolo Rech. 2019. Kernel and layer vulnerability factor to evaluate object detection reliability in GPUs. *IET Computers and Digital Techniques* 13, 3 (2019), 178–186.
- [240] P. Sarathy et al. 2019. Realizing the promise of artificial intelligence for unmanned aircraft systems through behavior bounded assurance. In *Proceedings of the IEEE/AIAA 38th Digital Avionics Systems Conference*. 1–8. DOI : <https://doi.org/10.1109/DASC43569.2019.9081649>
- [241] Sebastian Schirmer et al. [n. d.]. Considerations of artificial intelligence safety engineering for unmanned aircraft. (*Computer Safety, Reliability, and Security*) (2018), 465–472.
- [242] Volker Schneider. 2021. *Artificial Intelligence and Functional Safety - A summary of the current challenges*. Report. TÜV SUD Rail GmbH. Retrieved from <https://metsta.fi/wp-content/uploads/2021/05/Artificial-Intelligence-and-Functional-Safety.pdf>
- [243] Catherine D. Schuman, Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. 2017. A Survey of Neuromorphic Computing and Neural Networks in Hardware. arXiv:1705.06963. Retrieved from <https://arxiv.org/abs/1705.06963>
- [244] Johann M. Ph Schumann and Yan Liu. 2010. *Applications of Neural Networks in High Assurance Systems*. Springer.
- [245] Gesina Schwalbe and Martin Schels. 2020. A survey on methods for the safety assurance of machine learning based systems. In *Proceedings of the 10th European Congress on Embedded Real Time Software and Systems*.
- [246] D. Sculley et al. 2015. Hidden technical debt in machine learning systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 2503–2511.

- [247] D. Serpanos, G. Ferrari, G. Nikolakopoulos, J. Perez, M. Tauber, and S. Van Baelen. 2020. Embedded artificial intelligence: The ARTEMIS vision. *Computer* 53, 11 (2020), 65–69. DOI : <https://doi.org/10.1109/MC.2020.3016104>
- [248] Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. 2018. Uncertainty in machine learning: A safety perspective on autonomous driving. In *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, AS-SURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings* 37. Springer International Publishing, 458–464.
- [249] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 1528–1540.
- [250] T. Sheridan and W. Verplank. 1978. *Human and Computer Control of Undersea Teleoperators*. Report. MIT.
- [251] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proceedings of the IEEE Symposium on Security and Privacy*. 3–18. DOI : <https://doi.org/10.1109/SP.2017.41>
- [252] Christophe Simon, Walid Mechri, and Guillaume Capizzi. 2019. Assessment of safety integrity level by simulation of dynamic bayesian networks considering test duration. *Journal of Loss Prevention in the Process Industries* 57 (2019), 101–113. DOI : <https://doi.org/10.1016/j.jlp.2018.11.002>
- [253] SPARC. 2016. *Robotics 2020 - Multi-Annual Roadmap For Robotics in Europe - Horizon 2020 Call ICT-2017 (ICT-25, ICT-27 & ICT-28)*. Report. SPARC (The Partnership for Robotics in Europe).
- [254] Chen Sun et al. 2019. Cross validation for CNN based affordance learning and control for autonomous driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*. 1519–1524. DOI : <https://doi.org/10.1109/ITSC.2019.8917385>
- [255] Xiaowu Sun, Haitham Khedr, and Yasser Shoukry. 2019. Formal verification of neural network controlled autonomous systems. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. 147–156.
- [256] H. Tabani et al. 2019. Assessing the adherence of an industrial autonomous driving framework to ISO 26262 software guidelines. In *Proceedings of the 56th ACM/IEEE Design Automation Conference*. 1–6.
- [257] Hamid Tabani, Roger Pujol, Jaume Abella, and Francisco J. Cazorla. 2020. A cross-layer review of deep learning frameworks to ease their optimization and reuse. In *Proceedings of the 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing*. 144–145. DOI : <https://doi.org/10.1109/ISORC49007.2020.00030>
- [258] E. Talpes et al. 2020. Compute solution for Tesla's full self-driving computer. *IEEE Micro* 40, 2 (2020), 25–35. DOI : <https://doi.org/10.1109/MM.2020.2975764>
- [259] Holger Täubig, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter. 2012. Guaranteeing functional safety: Design for provability and computer-aided verification. *Autonomous Robots* 32, 3 (2012), 303–331.
- [260] Brian Taylor, Marjorie Darrah, and Christina Moats. 2003. Verification and validation of neural networks: A sampling of research in progress. In *Proceedings of SPIE - The International Society for Optical Engineering*. DOI : <https://doi.org/10.1117/12.487527>
- [261] Brian J. Taylor. 2006. *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. Springer Science & Business Media.
- [262] Brian J. Taylor, Marjorie A. Darrah, and Christina D. Moats. 2003. Verification and validation of neural networks: A sampling of research in progress. In *Proceedings of the SPIE 5103, Intelligent Computing: Theory and Applications*. 8–16. DOI : <https://doi.org/10.1117/12.487527>
- [263] Francesco Terrosi, Lorenzo Stringini, and Andrea Bondavalli. [n. d.]. Impact of machine learning on safety monitors. In *Proceedings of the Computer Safety, Reliability, and Security*. Mario Trapp, Francesca Saglietti, Marc Spisländer, and Friedemann Bitsch (Eds.), Springer Int. Publishing, 129–143.
- [264] N. Theuretzbacher. 1987. ELEKTRA: A system architecture that applies new principles to electronic interlocking. *IFAC Proceedings Volumes* 20, 3 (1987), 329–336. DOI : [https://doi.org/10.1016/S1474-6670\(17\)55918-2](https://doi.org/10.1016/S1474-6670(17)55918-2)
- [265] Stephen Thomas and Dirk Vandenberg. 2019. Harnessing uncertainty in autonomous vehicle safety. *Journal of System Safety* 55, 2 (2019), 23–29. DOI : <https://doi.org/10.56094/jss.v55i2.46>
- [266] Miles S. Thompson. 2008. Testing the intelligence of unmanned autonomous systems. *ITEA Journal* 29 (2008), 380–387.
- [267] Risto Tiusanen, Timo Malm, and Ari Ronkainen. 2020. An overview of current safety requirements for autonomous machinesreview of standards. *Open Engineering* 10, 1 (2020), 665–673.
- [268] Christoph Torens, Franz Juenger, Sebastian Schirmer, Simon Schopferer, Theresa D. Maienschein, and Johann C. Dauer. 2022. Machine learning verification and safety for unmanned aircraft - A literature study. In *AIAA SCITECH 2022 Forum*. DOI : <https://doi.org/10.2514/6.2022-1133>

- [269] John Törnblom and Simin Nadjm-Tehrani. 2018. Formal verification of random forests in safety-critical applications. In *Proceedings of the International Workshop on Formal Techniques for Safety-Critical Systems*. Springer, 55–71.
- [270] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium*. 601–618.
- [271] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 8011–8021.
- [272] Ignacio Trojaola, Iker Elorza, Eloy Irigoyen, Aron Pujana-Arrese, and Carlos Calleja. 2020. Iterative learning control for a hydraulic cushion. In *Proceedings of the 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019) Seville, Spain, May 1315, 2019, Proceedings 14*. 503–512. DOI : https://doi.org/10.1007/978-3-030-20055-8_48
- [273] C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, and J. Kapinski. 2020. Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles* 5, 2 (2020), 265–280. DOI : <https://doi.org/10.1109/TIV.2019.2955903>
- [274] TUVR. 2022. *Basics of Machine Learning with Aspects of Functional Safety and Cybersecurity*. Report. TÜV Rheinland.
- [275] ULSE. 2020. UL 4600 - Standard for Evaluation of Autonomous Products.
- [276] Rakshith Varadaraju. 2007. *A Survey of Introducing Artificial Intelligence Into the Safety Critical System Software Design Process*. Report. University of Northern Iowa.
- [277] K. R. Varshney. 2016. Engineering safety in machine learning. In *Proceedings of the Inform. Theory and Applicat. Workshop*. DOI : <https://doi.org/10.1109/ITA.2016.7888195>
- [278] Emil Vassev. 2016. Safe artificial intelligence and formal methods. In *Proceedings of the Leveraging Applicat. of Formal Methods, Verification and Validation: Foundational Techniques*. Tiziana Margaria and Bernhard Steffen (Eds.), Springer Internat. Publishing, 704–713.
- [279] Paulius Čerka et al. 2015. Liability for damages caused by artificial intelligence. *Computer Law & Security Review* 31, 3 (2015), 376–389. DOI : <https://doi.org/10.1016/j.clsr.2015.03.008>
- [280] VDE. 2021. VDE-AR-E 2842-61: Development and trustworthiness of autonomous/cognitive systems.
- [281] W. Wang and D. Zhao. 2018. Extracting traffic primitives directly from naturalistically logged data for self-driving applications. *IEEE Robotics and Automation Letters* 3, 2 (2018), 1223–1229. DOI : <https://doi.org/10.1109/LRA.2018.2794604>
- [282] Francis Rhys Ward and Ibrahim Habli. 2020. An assurance case pattern for the interpretability of machine learning in safety-critical systems. *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*. Springer Internat. Publishing, 395–407.
- [283] Waymo. 2019. AutoML: Automating the design of machine learning models for autonomous driving. Retrieved from <https://blog.waymo.com/2019/07/automl-automating-design-of-machine.html>
- [284] L. G. Weiss. 2011. Autonomous robots in the fog of war. *IEEE Spectrum* 48, 8 (2011), 30–57. DOI : <https://doi.org/10.1109/MSPEC.2011.5960163>
- [285] William H. Widen and Philip Koopman. 2022. Autonomous vehicle regulation & trust: Impact Of failures to comply with standards. *Journal of Law & Technology* 27, 3 (2022), 169–261. DOI : <https://doi.org/10.2139/ssrn.3969214>
- [286] Nan Wu and Yuan Xie. 2022. A survey of machine learning for computer architecture and systems. *ACM Computings Surveys*. 55, 3 (2022), 1–39. DOI : <https://doi.org/10.1145/3494523>
- [287] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature Squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings of the Network and Distributed System Security Symposium*.
- [288] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. 2020. ML-LOO: Detecting adversarial examples with feature attribution. *Proceedings of the AAAI Conference on Artificial Intelligence*. 6639–6647.
- [289] Junko Yoshida. 2020. *Unveiled: BMW's Scalable AV Architecture*. IEEE.
- [290] Katsuba Yurii and Grigorieva Liudmila. 2017. Application of artificial neural networks in vehicles' design self-diagnostic systems for safety reasons. *Transportation Research Procedia* 20 (2017), 283–287. DOI : <https://doi.org/10.1016/j.trpro.2017.01.024>
- [291] Jin Zhang and Jingyue Li. 2020. Testing and verification of neural-network-based safety-critical control software: A systematic literature review. *Inform. and Software Technology* 123 (2020), 106296. DOI : <https://doi.org/10.1016/j.infsof.2020.106296>
- [292] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* 48, 1 (2020), 1–36.

- [293] Ding Zhao, Xianan Huang, Huei Peng, Henry Lam, and David J. LeBlanc. 2018. Accelerated evaluation of automated vehicles in car-following maneuvers. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (2018), 733–744.
DOI:<https://doi.org/10.1109/TITS.2017.2701846>
- [294] Q. Zhu et al. 2021. Safety-assured design and adaptation of learning-enabled autonomous systems. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*. 753–760.

Received 13 September 2022; revised 14 July 2023; accepted 26 September 2023